Red Hat Enterprise Linux 4

Introduction To System Administration

For Red Hat Enterprise Linux 4

Red Hat Enterprise Linux 4 Introduction To System Administration
For Red Hat Enterprise Linux 4
Edition 2

1801 Varsity Drive
Raleigh, NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701

This book contains basic information about Red Hat Enterprise Linux system administration, and is suitable for system administrators with limited Linux experience.

Introduction

Welcome to the *Red Hat Enterprise Linux Introduction to System Adminitration*.

The *Red Hat Enterprise Linux Introduction to System Adminitration* contains introductory information for new Red Hat Enterprise Linux system administrators. It does *not* teach you how to perform a particular task under Red Hat Enterprise Linux; rather, it provides you with the background knowledge that more experienced system administrators have learned over time.

This guide assumes you have a limited amount of experience as a Linux user, but no Linux system administration experience. If you are completely new to Linux in general (and Red Hat Enterprise Linux in particular), you should start by purchasing an introductory book on Linux.

Each chapter in the *Red Hat Enterprise Linux Introduction to System Adminitration* has the following structure:

- Generic overview material -- This section discusses the topic of the chapter without going into details about a specific operating system, technology, or methodology.

- Red Hat Enterprise Linux-specific material -- This section addresses aspects of the topic related to Linux in general and Red Hat Enterprise Linux in particular.

- Additional resources for further study -- This section includes pointers to other Red Hat Enterprise Linux manuals, helpful websites, and books containing information applicable to the topic.

By adopting a consistent structure, readers can more easily read the *Red Hat Enterprise Linux Introduction to System Adminitration* in whatever way they choose. For example, an experienced system administrator with little Red Hat Enterprise Linux experience could skim only the sections that specifically focus on Red Hat Enterprise Linux, while a new system adminstrator could start by reading only the generic overview sections, and using the Red Hat Enterprise Linux-specific sections as an introduction to more in-depth resources.

While on the subject of more in-depth resources, the *System Administrators Guide* is an excellent resource for performing specific tasks in a Red Hat Enterprise Linux environment. Administrators requiring more in-depth, factual information should refer to the *Reference Guide*.

HTML, PDF, and RPM versions of the manuals are available on the Red Hat Enterprise Linux Documentation CD and online at *http://www.redhat.com/docs/manuals/enterprise/*.

1. Architecture-specific Information

Unless otherwise noted, all information contained in this manual apply only to the x86 processor and processors featuring the Intel® Extended Memory 64 Technology (Intel® EM64T) and AMD64 technologies. For architecture-specific information, refer to the *Red Hat Enterprise Linux Installation Guide* for your respective architecture.

2. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts*[1] set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not,

[1] https://fedorahosted.org/liberation-fonts/

alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

2.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to the first virtual terminal. Press **`Ctrl`**+**`Alt`**+**`F1`** to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh** ***username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.
>
> The **mount -o remount** ***file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.
>
> To see the version of a currently installed package, use the **rpm -q** ***package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above -- username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

2.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

2.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

3. More to Come

The *Red Hat Enterprise Linux Introduction to System Adminitration* is part of Red Hat, Inc's growing commitment to provide useful and timely support to Red Hat Enterprise Linux users. As new releases of Red Hat Enterprise Linux are made available, we make every effort to include both new and improved documentation for you.

3.1. Send in Your Feedback

If you spot a typo in the *Red Hat Enterprise Linux Introduction to System Adminitration*, or if you have thought of a way to make this manual better, we would love to hear from you. Please submit a report in Bugzilla (*http://bugzilla.redhat.com/bugzilla*[2]) against the component **`rhel-isa`**.

If you mention this manual's identifier, we will know exactly which version of the guide you have.

If you have a suggestion for improving the documentation, try to be as specific as possible. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

[2] http://bugzilla.redhat.com/bugzilla/

The Philosophy of System Administration

Although the specifics of being a system administrator may change from platform to platform, there are underlying themes that do not. These themes make up the philosophy of system administration.

The themes are:

- Automate everything
- Document everything
- Communicate as much as possible
- Know your resources
- Know your users
- Know your business
- Security cannot be an afterthought
- Plan ahead
- Expect the unexpected

The following sections explore each theme in more detail.

1.1. Automate Everything

Most system administrators are outnumbered -- either by their users, their systems, or both. In many cases, automation is the only way to keep up. In general, anything done more than once should be examined as a possible candidate for automation.

Here are some commonly automated tasks:

- Free disk space checking and reporting
- Backups
- System performance data collection
- User account maintenance (creation, deletion, etc.)
- Business-specific functions (pushing new data to a Web server, running monthly/quarterly/yearly reports, etc.)

This list is by no means complete; the functions automated by system administrators are only limited by an administrator's willingness to write the necessary scripts. In this case, being lazy (and making the computer do more of the mundane work) is actually a good thing.

Automation also gives users the extra benefit of greater predictability and consistency of service.

Note

Keep in mind that if you have a task that should be automated, it is likely that you are not the first system administrator to have that need. Here is where the benefits of open source software really shine -- you may be able to leverage someone else's work to automate the manual procedure that is currently eating up your time. So always make sure you search the Web before writing anything more complex than a small Perl script.

1.2. Document Everything

If given the choice between installing a brand-new server and writing a procedural document on performing system backups, the average system administrator would install the new server every time. While this is not at all unusual, you *must* document what you do. Many system administrators put off doing the necessary documentation for a variety of reasons:

"I will get around to it later."

Unfortunately, this is usually not true. Even if a system administrator is not kidding themselves, the nature of the job is such that everyday tasks are usually too chaotic to "do it later." Even worse, the longer it is put off, the more that is forgotten, leading to a much less detailed (and therefore, less useful) document.

"Why write it up? I will remember it."

Unless you are one of those rare individuals with a photographic memory, no, you will not remember it. Or worse, you will remember only half of it, not realizing that you are missing the whole story. This leads to wasted time either trying to relearn what you had forgotten or fixing what you had broken due to your incomplete understanding of the situation.

"If I keep it in my head, they will not fire me -- I will have job security!"

While this may work for a while, invariably it leads to less -- not more -- job security. Think for a moment about what may happen during an emergency. You may not be available; your documentation may save the day by letting someone else resolve the problem in your absence. And never forget that emergencies tend to be times when upper management pays close attention. In such cases, it is better to have your documentation be part of the solution than it is for your absence to be part of the problem.

In addition, if you are part of a small but growing organization, eventually there will be a need for another system administrator. How can this person learn to back you up if everything is in your head? Worst yet, not documenting may make you so indispensable that you might not be able to advance your career. You could end up working for the very person that was hired to assist you.

Hopefully you are now sold on the benefits of system documentation. That brings us to the next question: What should you document? Here is a partial list:

Policies

Policies are written to formalize and clarify the relationship you have with your user community. They make it clear to your users how their requests for resources and/or assistance are handled. The nature, style, and method of disseminating policies to your a community varies from organization to organization.

Procedures

Procedures are any step-by-step sequence of actions that must be taken to accomplish a certain task. Procedures to be documented can include backup procedures, user account management procedures, problem reporting procedures, and so on. Like automation, if a procedure is followed more than once, it is a good idea to document it.

Changes

A large part of a system administrator's career revolves around making changes -- configuring systems for maximum performance, tweaking scripts, modifying configuration files, and so on. All of these changes should be documented in some fashion. Otherwise, you could find yourself being completely confused about a change you made several months earlier.

Some organizations use more complex methods for keeping track of changes, but in many cases a simple revision history at the start of the file being changed is all that is necessary. At a minimum, each entry in the revision history should contain:

- The name or initials of the person making the change
- The date the change was made
- The reason the change was made

This results in concise, yet useful entries:

ECB, 12-June-2002 -- Updated entry for new Accounting printer (to support the replacement printer's ability to print duplex)

1.3. Communicate as Much as Possible

When it comes to your users, you can never communicate too much. Be aware that small system changes you might think are practically unnoticeable could very well completely confuse the administrative assistant in Human Resources.

The method by which you communicate with your users can vary according to your organization. Some organizations use email; others, an internal website. Still others may rely on Usenet news or IRC. A sheet of paper tacked to a bulletin board in the breakroom may even suffice at some places. In any case, use whatever method(s) that work well at your organization.

In general, it is best to follow this paraphrased approach used in writing newspaper stories:

1. Tell your users what you are going to do
2. Tell your users what you are doing
3. Tell your users what you have done

The following sections look at these steps in more depth.

1.3.1. Tell Your Users What You Are Going to Do

Make sure you give your users sufficient warning before you do anything. The actual amount of warning necessary varies according to the type of change (upgrading an operating system demands more lead time than changing the default color of the system login screen), as well as the nature of your user community (more technically adept users may be able to handle changes more readily than users with minimal technical skills.)

At a minimum, you should describe:

- The nature of the change
- When it will take place
- Why it is happening

- Approximately how long it should take
- The impact (if any) that the users can expect due to the change
- Contact information should they have any questions or concerns

Here is a hypothetical situation. The Finance department has been experiencing problems with their database server being very slow at times. You are going to bring the server down, upgrade the CPU module to a faster model, and reboot. Once this is done, you will move the database itself to faster, RAID-based storage. Here is one possible announcement for this situation:

> ### System Downtime Scheduled for Friday Night
>
> Starting this Friday at 6pm (midnight for our associates in Berlin), all financial applications will be unavailable for a period of approximately four hours.
>
> During this time, changes to both the hardware and software on the Finance database server will be performed. These changes should greatly reduce the time required to run the Accounts Payable and Accounts Receivable applications, and the weekly Balance Sheet report.
>
> Other than the change in runtime, most people should notice no other change. However, those of you that have written your own SQL queries should be aware that the layout of some indices will change. This is documented on the company intranet website, on the Finance page.
>
> Should you have any questions, comments, or concerns, please contact System Administration at extension 4321.

A few points are worth noting:

- Effectively communicate the start and duration of any downtime that might be involved in the change.
- Make sure you give the time of the change in such a way that it is useful to *all* users, no matter where they may be located.
- Use terms that your users understand. The people impacted by this work do not care that the new CPU module is a 2GHz unit with twice as much L2 cache, or that the database is being placed on a RAID 5 logical volume.

1.3.2. Tell Your Users What You Are Doing

This step is primarily a last-minute warning of the impending change; as such, it should be a brief repeat of the first message, though with the impending nature of the change made more apparent ("The system upgrade will take place TONIGHT."). This is also a good place to publicly answer any questions you may have received as a result of the first message.

Continuing our hypothetical example, here is one possible last-minute warning:

> ### System Downtime Scheduled for *Tonight*
>
> Reminder: The system downtime announced this past Monday will take place as scheduled tonight at 6pm (midnight for the Berlin office). You can find the original announcement on the company intranet website, on the System Administration page.
>
> Several people have asked whether they should stop working early tonight to make sure their work is backed up prior to the downtime. This will not be necessary, as the work being done tonight will not impact any work done on your personal workstations.

> Remember, those of you that have written your own SQL queries should be aware that the layout of some indices will change. This is documented on the company intranet website, on the Finance page.

Your users have been alerted; now you are ready to actually do the work.

1.3.3. Tell Your Users What You Have Done

After you have finished making the changes, you *must* tell your users what you have done. Again, this should be a summary of the previous messages (invariably someone will not have read them.)[1]

However, there is one important addition you must make. It is vital that you give your users the current status. Did the upgrade not go as smoothly as planned? Was the new storage server only able to serve the systems in Engineering, and not in Finance? These types of issues must be addressed here.

Of course, if the current status differs from what you communicated previously, you should make this point clear and describe what will be done (if anything) to arrive at the final solution.

In our hypothetical situation, the downtime had some problems. The new CPU module did not work; a call to the system's manufacturer revealed that a special version of the module is required for in-the-field upgrades. On the plus side, the migration of the database to the RAID volume went well (even though it took a bit longer than planned due to the problems with the CPU module.

Here is one possible announcement:

> **System Downtime Complete**
>
> The system downtime scheduled for Friday night (refer to the System Administration page on the company intranet website) has been completed. Unfortunately, hardware issues prevented one of the tasks from being completed. Due to this, the remaining tasks took longer than the originally-scheduled four hours. Instead, all systems were back in production by midnight (6am Saturday for the Berlin office).
>
> Because of the remaining hardware issues, performance of the AP, AR, and the Balance Sheet report will be slightly improved, but not to the extent originally planned. A second downtime will be announced and scheduled as soon as the issues that prevented completion of the task have been resolved.
>
> Please note that the downtime did change some database indices; people that have written their own SQL queries should consult the Finance page on the company intranet website.
>
> Please contact System Administration at extension 4321 with any questions.

With this kind of information, your users will have sufficient background knowledge to continue their work, and to understand how the changes impact them.

1.4. Know Your Resources

System administration is mostly a matter of balancing available resources against the people and programs that use those resources. Therefore, your career as a system administrator will be a short and stress-filled one unless you fully understand the resources you have at your disposal.

[1] Be sure to send this message out as soon as the work is done, *before* you leave for home. Once you have left the office, it is much too easy to forget, leaving your users in the dark as to whether they can use the system or not.

Some of the resources are ones that seem pretty obvious:

- System resources, such as available processing power, memory, and disk space
- Network bandwidth
- Available money in the IT budget

But some may not be so obvious:

- The services of operations personnel, other system administrators, or even an administrative assistant
- Time (often of critical importance when the time involves things such as the amount of time during which system backups may take place)
- Knowledge (whether it is stored in books, system documentation, or the brain of a person that has worked at the company for the past twenty years)

It is important to note is that it is highly valuable to take a complete inventory of those resources available to you and to *keep it current* -- a lack of "situational awareness" when it comes to available resources can often be worse than *no* awareness at all.

1.5. Know Your Users

Although some people bristle at the term "users" (perhaps due to some system administrators' use of the term in a derogatory manner), it is used here with no such connotation implied. Users are those people that use the systems and resources for which you are responsible -- no more, and no less. As such, they are central to your ability to successfully administer your systems; without understanding your users, how can you understand the system resources they require?

For example, consider a bank teller. A bank teller uses a strictly-defined set of applications and requires little in the way of system resources. A software engineer, on the other hand, may use many different applications and always welcomes more system resources (for faster build times). Two entirely different users with two entirely different needs.

Make sure you learn as much about your users as you can.

1.6. Know Your Business

Whether you work for a large, multinational corporation or a small community college, you must still understand the nature of the business environment in which you work. This can be boiled down to one question:

What is the purpose of the systems you administer?

The key point here is to understand your systems' purpose in a more global sense:

- Applications that must be run within certain time frames, such as at the end of a month, quarter, or year
- The times during which system maintenance may be done
- New technologies that could be used to resolve long-standing business problems

By taking into account your organization's business, you will find that your day-to-day decisions will be better for your users, and for you.

1.7. Security Cannot be an Afterthought

No matter what you might think about the environment in which your systems are running, you cannot take security for granted. Even standalone systems not connected to the Internet may be at risk (although obviously the risks will be different from a system that has connections to the outside world).

Therefore, it is extremely important to consider the security implications of everything you do. The following list illustrates the different kinds of issues you should consider:

- The nature of possible threats to each of the systems under your care
- The location, type, and value of the data on those systems
- The type and frequency of authorized access to the systems

While you are thinking about security, do not make the mistake of assuming that possible intruders will only attack your systems from outside of your company. Many times the perpetrator is someone within the company. So the next time you walk around the office, look at the people around you and ask yourself this question:

What would happen if *that* person were to attempt to subvert our security?

Note

This does *not* mean that you should treat your coworkers as if they are criminals. It just means that you should look at the type of work that each person performs and determine what types of security breaches a person in that position could perpetrate, if they were so inclined.

1.7.1. The Risks of Social Engineering

While most system administrators' first reactions when they think about security is to concentrate on the technological aspects, it is important to maintain perspective. Quite often, security breaches do not have their origins in technology, but in human nature.

People interested in breaching security often use human nature to entirely bypass technological access controls. This is known as *social engineering*. Here is an example:

The second shift operator receives an outside phone call. The caller claims to be your organization's CFO (the CFO's name and background information was obtained from your organization's website, on the "Management Team" page).

The caller claims to be calling from some place halfway around the world (maybe this part of the story is a complete fabrication, or perhaps your organization's website has a recent press release that makes mention of the CFO attending a tradeshow).

The caller tells a tale of woe; his laptop was stolen at the airport, and he is with an important customer and needs access to the corporate intranet to check on the customer's account status. Would the operator be so kind as to give him the necessary access information?

Do you know what would your operator do? Unless your operator has guidance (in the form of policies and procedures), you very likely do not know for sure.

Like traffic lights, the goal of policies and procedures is to provide unambiguous guidance as to what is and is not appropriate behavior. However, just as with traffic lights, policies and procedures only work if everyone follows them. And there is the crux of the problem -- it is unlikely that everyone will adhere to your policies and procedures. In fact, depending on the nature of your organization, it is possible that you do not even have sufficient authority to define policies, much less enforce them. What then?

Unfortunately, there are no easy answers. User education can help; do everything you can to help make your user community aware of security and social engineering. Give lunchtime presentations about security. Post pointers to security-related news articles on your organization's mailing lists. Make yourself available as a sounding board for users' questions about things that do not seem quite right.

In short, get the message out to your users any way you can.

1.8. Plan Ahead

System administrators that took all this advice to heart and did their best to follow it would be fantastic system administrators -- for a day. Eventually, the environment will change, and one day our fantastic administrator would be caught flat-footed. The reason? Our fantastic administrator failed to plan ahead.

Certainly no one can predict the future with 100% accuracy. However, with a bit of awareness it is easy to read the signs of many changes:

- An offhand mention of a new project gearing up during that boring weekly staff meeting is a sure sign that you will likely need to support new users in the near future

- Talk of an impending acquisition means that you may end up being responsible for new (and possibly incompatible) systems in one or more remote locations

Being able to read these signs (and to respond effectively to them) makes life easier for you and your users.

1.9. Expect the Unexpected

While the phrase "expect the unexpected" is trite, it reflects an underlying truth that all system administrators must understand:

There *will* be times when you are caught off-guard.

After becoming comfortable with this uncomfortable fact of life, what can a concerned system administrator do? The answer lies in flexibility; by performing your job in such a way as to give you (and your users) the most options possible. Take, for example, the issue of disk space. Given that never having sufficient disk space seems to be as much a physical law as the law of gravity, it is reasonable to assume that at some point you will be confronted with a desperate need for additional disk space *right now*.

What would a system administrator who expects the unexpected do in this case? Perhaps it is possible to keep a few disk drives sitting on the shelf as spares in case of hardware problems[2]. A spare of this type could be quickly deployed[3] on a temporary basis to address the short-term need for disk space, giving time to more permanently resolve the issue (by following the standard procedure for procuring additional disk drives, for example).

By trying to anticipate problems before they occur, you will be in a position to respond more quickly and effectively than if you let yourself be surprised.

[2] And of course a system administrator that expects the unexpected would naturally use RAID (or related technologies) to lessen the impact of a critical disk drive failing during production.

[3] Again, system administrators that think ahead configure their systems to make it as easy as possible to quickly add a new disk drive to the system.

1.10. Red Hat Enterprise Linux-Specific Information

This section describes information related to the philosophy of system administration that is specific to Red Hat Enterprise Linux.

1.10.1. Automation

Automation of frequently-performed tasks under Red Hat Enterprise Linux requires knowledge of several different types of technologies. First are the commands that control the timing of command or script execution. The **`cron`** and **`at`** commands are most commonly used in these roles.

Incorporating an easy-to-understand yet powerfully flexible time specification system, **`cron`** can schedule the execution of commands or scripts for recurring intervals ranging in length from minutes to months. The **`crontab`** command is used to manipulate the files controlling the **`cron`** daemon that actually schedules each **`cron`** job for execution.

The **`at`** command (and the closely-related command **`batch`**) are more appropriate for scheduling the execution of one-time scripts or commands. These commands implement a rudimentary batch subsystem consisting of multiple queues with varying scheduling priorities. The priorities are known as *niceness* levels (due to the name of the command -- **`nice`**). Both **`at`** and **`batch`** are perfect for tasks that must start at a given time but are not time-critical in terms of finishing.

Next are the various scripting languages. These are the "programming languages" that the average system administrator uses to automate manual operations. There are many scripting languages (and each system administrator tends to have a personal favorite), but the following are currently the most common:

- The **`bash`** command shell
- The **`perl`** scripting language
- The **`python`** scripting language

Over and above the obvious differences between these languages, the biggest difference is in the way in which these languages interact with other utility programs on a Red Hat Enterprise Linux system. Scripts written with the **`bash`** shell tend to make more extensive use of the many small utility programs (for example, to perform character string manipulation), while **`perl`** scripts perform more of these types of operations using features built into the language itself. A script written using **`python`** can fully exploit the language's object-oriented capabilities, making complex scripts more easily extensible.

This means that, in order to truly master shell scripting, you must be familiar with the many utility programs (such as **`grep`** and **`sed`**) that are part of Red Hat Enterprise Linux. Learning **`perl`** (and **`python`**), on the other hand, tends to be a more "self-contained" process. However, many **`perl`** language constructs are based on the syntax of various traditional UNIX utility programs, and as such are familiar to those Red Hat Enterprise Linux system administrators with shell scripting experience.

1.10.2. Documentation and Communication

In the areas of documentation and communication, there is little that is specific to Red Hat Enterprise Linux. Since documentation and communication can consist of anything from adding comments to a text-based configuration file to updating a webpage or sending an email, a system administrator using Red Hat Enterprise Linux must have access to text editors, HTML editors, and mail clients.

Here is a small sample of the many text editors available under Red Hat Enterprise Linux:

- The **gedit** text editor
- The **Emacs** text editor

- The **`Vim`** text editor

The **gedit** text editor is a strictly graphical application (in other words, it requires an active X Window System environment), while **`vim`** and **Emacs** are primarily text-based in nature.

The subject of the best text editor has sparked debate for nearly as long as computers have existed and will continue to do so. Therefore, the best approach is to try each editor for yourself, and use what works best for you.

For HTML editors, system administrators can use the Composer function of the **Mozilla** Web browser. Of course, some system administrators prefer to hand-code their HTML, making a regular text editor a perfectly acceptable tool as well.

As far as email is concerned, Red Hat Enterprise Linux includes the **Evolution** graphical email client, the **Mozilla** email client (which is also graphical), and **mutt**, which is text-based. As with text editors, the choice of an email client tends to be a personal one; therefore, the best approach is to try each client for yourself, and use what works best for you.

1.10.3. Security

As stated earlier in this chapter, security cannot be an afterthought, and security under Red Hat Enterprise Linux is more than skin-deep. Authentication and access controls are deeply-integrated into the operating system and are based on designs gleaned from long experience in the UNIX community.

For authentication, Red Hat Enterprise Linux uses PAM -- Pluggable Authentication Modules. PAM makes it possible to fine-tune user authentication via the configuration of shared libraries that all PAM-aware applications use, all without requiring any changes to the applications themselves.

Access control under Red Hat Enterprise Linux uses traditional UNIX-style permissions (read, write, execute) against user, group, and "everyone else" classifications. Like UNIX, Red Hat Enterprise Linux also makes use of *setuid* and *setgid* bits to temporarily confer expanded access rights to processes running a particular program, based on the ownership of the program file. Of course, this makes it critical that any program to be run with setuid or setgid privileges must be carefully audited to ensure that no exploitable vulnerabilities exist.

Red Hat Enterprise Linux also includes support for *access control lists*. An access control list (ACL) is a construct that allows extremely fine-grained control over what users or groups may access a file or directory. For example, a file's permissions may restrict all access by anyone other than the file's owner, yet the file's ACL can be configured to allow only user **bob** to write and group **`finance`** to read the file.

Another aspect of security is being able to keep track of system activity. Red Hat Enterprise Linux makes extensive use of logging, both at a kernel and an application level. Logging is controlled by the system logging daemon **`syslogd`**, which can log system information locally (normally to files in the **`/var/log/`** directory) or to a remote system (which acts as a dedicated log server for multiple computers.)

Intrusion detection sytems (IDS) are powerful tools for any Red Hat Enterprise Linux system administrator. An IDS makes it possible for system administrators to determine whether unauthorized changes were made to one or more systems. The overall design of the operating system itself includes IDS-like functionality.

Because Red Hat Enterprise Linux is installed using the RPM Package Manager (RPM), it is possible to use RPM to verify whether any changes have been made to the packages comprising the operating system. However, because RPM is primarily a package management tool, its abilities as an IDS are somewhat limited. Even so, it can be a good first step toward monitoring a Red Hat Enterprise Linux system for unauthorized modifications.

1.11. Additional Resources

This section includes various resources that can be used to learn more about the philosophy of system administration and the Red Hat Enterprise Linux-specific subject matter discussed in this chapter.

1.11.1. Installed Documentation

The following resources are installed in the course of a typical Red Hat Enterprise Linux installation and can help you learn more about the subject matter discussed in this chapter.

- **`crontab(1)`** and **`crontab(5)`** man pages -- Learn how to schedule commands and scripts for automatic execution at regular intervals.
- **`at(1)`** man page -- Learn how to schedule commands and scripts for execution at a later time.
- **`bash(1)`** man page -- Learn more about the default shell and shell script writing.
- **`perl(1)`** man page -- Review pointers to the many man pages that make up perl's online documentation.
- **`python(1)`** man page -- Learn more about options, files, and environment variables controlling the Python interpreter.
- **`gedit(1)`** man page and **Help** menu entry -- Learn how to edit text files with this graphical text editor.
- **`emacs(1)`** man page -- Learn more about this highly-flexible text editor, including how to run its online tutorial.
- **`vim(1)`** man page -- Learn how to use this powerful text editor.
- **Mozilla Help Contents** menu entry -- Learn how to edit HTML files, read mail, and browse the Web.
- **`evolution(1)`** man page and **Help** menu entry -- Learn how to manage your email with this graphical email client.
- **`mutt(1)`** man page and files in **`/usr/share/doc/mutt-<version>`** -- Learn how to manage your email with this text-based email client.
- **`pam(8)`** man page and files in **`/usr/share/doc/pam-<version>`** -- Learn how authentication takes place under Red Hat Enterprise Linux.

1.11.2. Useful Websites

- *http://www.kernel.org/pub/linux/libs/pam/* -- The Linux-PAM project homepage.
- *http://www.usenix.org/* -- The USENIX homepage. A professional organization dedicated to bringing together computer professionals of all types and fostering improved communication and innovation.
- *http://www.sage.org/* -- The System Administrators Guild homepage. A USENIX special technical group that is a good resource for all system administrators responsible for Linux (or Linux-like) operating systems.
- *http://www.python.org/* -- The Python Language Website. An excellent site for learning more about Python.
- *http://www.perl.org/* -- The Perl Mongers Website. A good place to start learning about Perl and connecting with the Perl community.

- *http://www.rpm.org/* -- The RPM Package Manager homepage. The most comprehensive website for learning about RPM.

1.11.3. Related Books

Most books on system administration do little to cover the philosophy behind the job. However, the following books do have sections that give a bit more depth to the issues that were discussed here:

- The *Reference Guide*; Red Hat, Inc -- Provides an overview of locations of key system files, user and group settings, and PAM configuration.
- The *Security Guide*; Red Hat, Inc -- Contains a comprehensive discussion of many security-related issues for Red Hat Enterprise Linux system administrators.
- The *System Administrators Guide*; Red Hat, Inc -- Includes chapters on managing users and groups, automating tasks, and managing log files.
- *Linux Administration Handbook* by Evi Nemeth, Garth Snyder, and Trent R. Hein; Prentice Hall -- Provides a good section on the policies and politics side of system administration, including several "what-if" discussions concerning ethics.
- *Linux System Administration: A User's Guide* by Marcel Gagne; Addison Wesley Professional -- Contains a good chapter on automating various tasks.
- *Solaris System Management* by John Philcox; New Riders Publishing -- Although not specifically written for Red Hat Enterprise Linux (or even Linux in general), and using the term "system manager" instead of "system administrator," this book provides a 70-page overview of the many roles that system administrators play in a typical organization.

Resource Monitoring

As stated earlier, a great deal of system administration revolves around resources and their efficient use. By balancing various resources against the people and programs that use those resources, you waste less money and make your users as happy as possible. However, this leaves two questions:

What are resources?

And:

How is it possible to know what resources are being used (and to what extent)?

The purpose of this chapter is to enable you to answer these questions by helping you to learn more about resources and how they can be monitored.

2.1. Basic Concepts

Before you can monitor resources, you first have to know what resources there are to monitor. All systems have the following resources available:

- CPU power
- Bandwidth
- Memory
- Storage

These resources are covered in more depth in the following chapters. However, for the time being all you need to keep in mind is that these resources have a direct impact on system performance, and therefore, on your users' productivity and happiness.

At its simplest, resource monitoring is nothing more than obtaining information concerning the utilization of one or more system resources.

However, it is rarely this simple. First, one must take into account the resources to be monitored. Then it is necessary to examine each system to be monitored, paying particular attention to each system's situation.

The systems you monitor fall into one of two categories:

- The system is currently experiencing performance problems at least part of the time and you would like to improve its performance.
- The system is currently running well and you would like it to stay that way.

The first category means you should monitor resources from a system performance perspective, while the second category means you should monitor system resources from a capacity planning perspective.

Because each perspective has its own unique requirements, the following sections explore each category in more depth.

2.2. System Performance Monitoring

As stated above, system performance monitoring is normally done in response to a performance problem. Either the system is running too slowly, or programs (and sometimes even the entire system) fail to run at all. In either case, performance monitoring is normally done as the first and last steps of a three-step process:

1. Monitoring to identify the nature and scope of the resource shortages that are causing the performance problems

2. The data produced from monitoring is analyzed and a course of action (normally performance tuning and/or the procurement of additional hardware) is taken to resolve the problem

3. Monitoring to ensure that the performance problem has been resolved

Because of this, performance monitoring tends to be relatively short-lived in duration and more detailed in scope.

Note

System performance monitoring is often an iterative process, with these steps being repeated several times to arrive at the best possible system performance. The primary reason for this is that system resources and their utilization tend to be highly interrelated, meaning that often the elimination of one resource bottleneck uncovers another one.

2.3. Monitoring System Capacity

Monitoring system capacity is done as part of an ongoing capacity planning program. Capacity planning uses long-term resource monitoring to determine rates of change in the utilization of system resources. Once these rates of change are known, it becomes possible to conduct more accurate long-term planning regarding the procurement of additional resources.

Monitoring done for capacity planning purposes is different from performance monitoring in two ways:

- The monitoring is done on a more-or-less continuous basis

- The monitoring is usually not as detailed

The reason for these differences stems from the goals of a capacity planning program. Capacity planning requires a "big picture" viewpoint; short-term or anomalous resource usage is of little concern. Instead, data is collected over a period of time, making it possible to categorize resource utilization in terms of changes in workload. In more narrowly-defined environments, (where only one application is run, for example) it is possible to model the application's impact on system resources. This can be done with sufficient accuracy to make it possible to determine, for example, the impact of five more customer service representatives running the customer service application during the busiest time of the day.

2.4. What to Monitor?

As stated earlier, the resources present in every system are CPU power, bandwidth, memory, and storage. At first glance, it would seem that monitoring would need only consist of examining these four different things.

Unfortunately, it is not that simple. For example, consider a disk drive. What things might you want to know about its performance?

- How much free space is available?

- How many I/O operations on average does it perform each second?

- How long on average does it take each I/O operation to be completed?
- How many of those I/O operations are reads? How many are writes?
- What is the average amount of data read/written with each I/O?

There are more ways of studying disk drive performance; these points have only scratched the surface. The main concept to keep in mind is that there are many different types of data for each resource.

The following sections explore the types of utilization information that would be helpful for each of the major resource types.

2.4.1. Monitoring CPU Power

In its most basic form, monitoring CPU power can be no more difficult than determining if CPU utilization ever reaches 100%. If CPU utilization stays below 100%, no matter what the system is doing, there is additional processing power available for more work.

However, it is a rare system that does not reach 100% CPU utilization at least some of the time. At that point it is important to examine more detailed CPU utilization data. By doing so, it becomes possible to start determining where the majority of your processing power is being consumed. Here are some of the more popular CPU utilization statistics:

User Versus System

> The percentage of time spent performing user-level processing versus system-level processing can point out whether a system's load is primarily due to running applications or due to operating system overhead. High user-level percentages tend to be good (assuming users are not experiencing unsatisfactory performance), while high system-level percentages tend to point toward problems that will require further investigation.

Context Switches

> A context switch happens when the CPU stops running one process and starts running another. Because each context switch requires the operating system to take control of the CPU, excessive context switches and high levels of system-level CPU consumption tend to go together.

Interrupts

> As the name implies, interrupts are situations where the processing being performed by the CPU is abruptly changed. Interrupts generally occur due to hardware activity (such as an I/O device completing an I/O operation) or due to software (such as software interrupts that control application processing). Because interrupts must be serviced at a system level, high interrupt rates lead to higher system-level CPU consumption.

Runnable Processes

> A process may be in different states. For example, it may be:
>
> - Waiting for an I/O operation to complete
> - Waiting for the memory management subsystem to handle a page fault
>
> In these cases, the process has no need for the CPU.
>
> However, eventually the process state changes, and the process becomes runnable. As the name implies, a runnable process is one that is capable of getting work done as soon as it is scheduled to receive CPU time. However, if more than one process is runnable at any given time, all but

one[1] of the runnable processes must wait for their turn at the CPU. By monitoring the number of runnable processes, it is possible to determine how CPU-bound your system is.

Other performance metrics that reflect an impact on CPU utilization tend to include different services the operating system provides to processes. They may include statistics on memory management, I/O processing, and so on. These statistics also reveal that, when system performance is monitored, there are no boundaries between the different statistics. In other words, CPU utilization statistics may end up pointing to a problem in the I/O subsystem, or memory utilization statistics may reveal an application design flaw.

Therefore, when monitoring system performance, it is not possible to examine any one statistic in complete isolation; only by examining the overall picture it it possible to extract meaningful information from any performance statistics you gather.

2.4.2. Monitoring Bandwidth

Monitoring bandwidth is more difficult than the other resources described here. The reason for this is due to the fact that performance statistics tend to be device-based, while most of the places where bandwidth is important tend to be the buses that connect devices. In those instances where more than one device shares a common bus, you might see reasonable statistics for each device, but the aggregate load those devices place on the bus would be much greater.

Another challenge to monitoring bandwidth is that there can be circumstances where statistics for the devices themselves may not be available. This is particularly true for system expansion buses and datapaths[2]. However, even though 100% accurate bandwidth-related statistics may not always be available, there is often enough information to make some level of analysis possible, particularly when related statistics are taken into account.

Some of the more common bandwidth-related statistics are:

Bytes received/sent

Network interface statistics provide an indication of the bandwidth utilization of one of the more visible buses -- the network.

Interface counts and rates

These network-related statistics can give indications of excessive collisions, transmit and receive errors, and more. Through the use of these statistics (particularly if the statistics are available for more than one system on your network), it is possible to perform a modicum of network troubleshooting even before the more common network diagnostic tools are used.

Transfers per Second

Normally collected for block I/O devices, such as disk and high-performance tape drives, this statistic is a good way of determining whether a particular device's bandwidth limit is being reached. Due to their electromechanical nature, disk and tape drives can only perform so many I/O operations every second; their performance degrades rapidly as this limit is reached.

2.4.3. Monitoring Memory

If there is one area where a wealth of performance statistics can be found, it is in the area of monitoring memory utilization. Due to the inherent complexity of today's demand-paged virtual memory operating systems, memory utilization statistics are many and varied. It is here that the majority of a system administrator's work with resource management takes place.

[1] Assuming a single-processor computer system.

[2] More information on buses, datapaths, and bandwidth is available in *Chapter 3, Bandwidth and Processing Power*.

The following statistics represent a cursory overview of commonly-found memory management statistics:

Page Ins/Page Outs
: These statistics make it possible to gauge the flow of pages from system memory to attached mass storage devices (usually disk drives). High rates for both of these statistics can mean that the system is short of physical memory and is *thrashing*, or spending more system resources on moving pages into and out of memory than on actually running applications.

Active/Inactive Pages
: These statistics show how heavily memory-resident pages are used. A lack of inactive pages can point toward a shortage of physical memory.

Free, Shared, Buffered, and Cached Pages
: These statistics provide additional detail over the more simplistic active/inactive page statistics. By using these statistics, it is possible to determine the overall mix of memory utilization.

Swap Ins/Swap Outs
: These statistics show the system's overall swapping behavior. Excessive rates here can point to physical memory shortages.

Successfully monitoring memory utilization requires a good understanding of how demand-paged virtual memory operating systems work. While such a subject alone could take up an entire book, the basic concepts are discussed in *Chapter 4, Physical and Virtual Memory*. This chapter, along with time spent actually monitoring a system, gives you the the necessary building blocks to learn more about this subject.

2.4.4. Monitoring Storage

Monitoring storage normally takes place at two different levels:

- Monitoring for sufficient disk space
- Monitoring for storage-related performance problems

The reason for this is that it is possible to have dire problems in one area and no problems whatsoever in the other. For example, it is possible to cause a disk drive to run out of disk space without once causing any kind of performance-related problems. Likewise, it is possible to have a disk drive that has 99% free space, yet is being pushed past its limits in terms of performance.

However, it is more likely that the average system experiences varying degrees of resource shortages in both areas. Because of this, it is also likely that -- to some extent -- problems in one area impact the other. Most often this type of interaction takes the form of poorer and poorer I/O performance as a disk drive nears 0% free space although, in cases of extreme I/O loads, it might be possible to slow I/O throughput to such a level that applications no longer run properly.

In any case, the following statistics are useful for monitoring storage:

Free Space
: Free space is probably the one resource all system administrators watch closely; it would be a rare administrator that never checks on free space (or has some automated way of doing so).

File System-Related Statistics
: These statistics (such as number of files/directories, average file size, etc.) provide additional detail over a single free space percentage. As such, these statistics make it possible for system administrators to configure the system to give the best performance, as the I/O load imposed by

a file system full of many small files is not the same as that imposed by a file system filled with a single massive file.

Transfers per Second

This statistic is a good way of determining whether a particular device's bandwidth limitations are being reached.

Reads/Writes per Second

A slightly more detailed breakdown of transfers per second, these statistics allow the system administrator to more fully understand the nature of the I/O loads a storage device is experiencing. This can be critical, as some storage technologies have widely different performance characteristics for read versus write operations.

2.5. Red Hat Enterprise Linux-Specific Information

Red Hat Enterprise Linux comes with a variety of resource monitoring tools. While there are more than those listed here, these tools are representative in terms of functionality. The tools are:

- **`free`**
- **`top`** (and **GNOME System Monitor**, a more graphically oriented version of **`top`**)
- **`vmstat`**
- The Sysstat suite of resource monitoring tools
- The OProfile system-wide profiler

Let us examine each one in more detail.

2.5.1. `free`

The **`free`** command displays system memory utilization. Here is an example of its output:

```
 total used free shared buffers cached Mem: 255508 240268 15240 0 7592 86188 -/+ buffers/
cache: 146488 109020 Swap: 530136 26268 503868
```

The **`Mem:`** row displays physical memory utilization, while the **`Swap:`** row displays the utilization of the system swap space, and the **`-/+ buffers/cache:`** row displays the amount of physical memory currently devoted to system buffers.

Since **`free`** by default only displays memory utilization information once, it is only useful for very short-term monitoring, or quickly determining if a memory-related problem is currently in progress. Although **`free`** has the ability to repetitively display memory utilization figures via its **`-s`** option, the output scrolls, making it difficult to easily detect changes in memory utilization.

Note

A better solution than using **`free -s`** would be to run **`free`** using the **`watch`** command. For example, to display memory utilization every two seconds (the default display interval for **`watch`**), use this command:

```
watch free
```

The **`watch`** command issues the **`free`** command every two seconds, updating by clearing the screen and writing the new output to the same screen location. This makes it much easier to determine how memory utilization changes over time, since **`watch`** creates a single updated view with no scrolling. You can control the delay between updates by using the **`-n`** option, and can cause any changes between updates to be highlighted by using the **`-d`** option, as in the following command:

```
watch -n 1 -d free
```

For more information, refer to the **`watch`** man page.

The **`watch`** command runs until interrupted with **`Ctrl`**+**`C`**. The **`watch`** command is something to keep in mind; it can come in handy in many situations.

2.5.2. `top`

While **`free`** displays only memory-related information, the **`top`** command does a little bit of everything. CPU utilization, process statistics, memory utilization -- **`top`** monitors it all. In addition, unlike the **`free`** command, **`top`**'s default behavior is to run continuously; there is no need to use the **`watch`** command. Here is a sample display:

```
14:06:32 up 4 days, 21:20, 4 users, load average: 0.00, 0.00, 0.00 77 processes: 76
sleeping, 1 running, 0 zombie, 0 stopped CPU states: cpu user nice system irq softirq iowait
idle total 19.6% 0.0% 0.0% 0.0% 0.0% 0.0% 180.2% cpu00 0.0% 0.0% 0.0% 0.0% 0.0% 0.0% 100.0%
cpu01 19.6% 0.0% 0.0% 0.0% 0.0% 0.0% 80.3% Mem: 1028548k av, 716604k used, 311944k free, 0k
shrd, 131056k buff 324996k actv, 108692k in_d, 13988k in_c Swap: 1020116k av, 5276k used,
1014840k free 382228k cached PID USER PRI NI SIZE RSS SHARE STAT %CPU %MEM TIME CPU COMMAND
17578 root 15 0 13456 13M 9020 S 18.5 1.3 26:35 1 rhn-applet-gu 19154 root 20 0 1176 1176
892 R 0.9 0.1 0:00 1 top 1 root 15 0 168 160 108 S 0.0 0.0 0:09 0 init 2 root RT 0 0 0 0 SW
0.0 0.0 0:00 0 migration/0 3 root RT 0 0 0 0 SW 0.0 0.0 0:00 1 migration/1 4 root 15 0 0 0 0
SW 0.0 0.0 0:00 0 keventd 5 root 34 19 0 0 0 SWN 0.0 0.0 0:00 0 ksoftirqd/0 6 root 35 19 0 0
0 SWN 0.0 0.0 0:00 1 ksoftirqd/1 9 root 15 0 0 0 0 SW 0.0 0.0 0:07 1 bdflush 7 root 15 0 0 0
0 SW 0.0 0.0 1:19 0 kswapd 8 root 15 0 0 0 0 SW 0.0 0.0 0:14 1 kscand 10 root 15 0 0 0 0 SW
0.0 0.0 0:03 1 kupdated 11 root 25 0 0 0 0 SW 0.0 0.0 0:00 0 mdrecoveryd
```

The display is divided into two sections. The top section contains information related to overall system status -- uptime, load average, process counts, CPU status, and utilization statistics for both memory and swap space. The lower section displays process-level statistics. It is possible to change what is displayed while **`top`** is running. For example, **`top`** by default displays both idle and non-idle processes. To display only non-idle processes, press **`i`**; a second press returns to the default display mode.

Warning

Although **top** appears like a simple display-only program, this is not the case. That is because **top** uses single character commands to perform various operations. For example, if you are logged in as root, it is possible to change the priority and even kill any process on your system. Therefore, until you have reviewed **top**'s help screen (type **?** to display it), it is safest to only type **q** (which exits **top**).

2.5.2.1. The GNOME System Monitor -- A Graphical `top`

If you are more comfortable with graphical user interfaces, the **GNOME System Monitor** may be more to your liking. Like **top**, the **GNOME System Monitor** displays information related to overall system status, process counts, memory and swap utilization, and process-level statistics.

However, the **GNOME System Monitor** goes a step further by also including graphical representations of CPU, memory, and swap utilization, along with a tabular disk space utilization listing. An example of the **GNOME System Monitor**'s **Process Listing** display appears in *Figure 2.1, “The **GNOME System Monitor Process Listing** Display”*.

Figure 2.1. The **GNOME System Monitor Process Listing** Display

Additional information can be displayed for a specific process by first clicking on the desired process and then clicking on the **More Info** button.

To display the CPU, memory, and disk usage statistics, click on the **System Monitor** tab.

2.5.3. `vmstat`

For a more concise understanding of system performance, try **vmstat**. With **vmstat**, it is possible to get an overview of process, memory, swap, I/O, system, and CPU activity in one line of numbers:

```
procs memory swap io system cpu r b swpd free buff cache si so bi bo in cs us sy id wa 0 0
5276 315000 130744 380184 1 1 2 24 14 50 1 1 47 0
```

The first line divides the fields in six categories, including process, memory, swap, I/O, system, and CPU related statistics. The second line further identifies the contents of each field, making it easy to quickly scan data for specific statistics.

The process-related fields are:

- **r** -- The number of runnable processes waiting for access to the CPU
- **b** -- The number of processes in an uninterruptible sleep state

The memory-related fields are:

- **swpd** -- The amount of virtual memory used
- **free** -- The amount of free memory
- **buff** -- The amount of memory used for buffers
- **cache** -- The amount of memory used as page cache

The swap-related fields are:

- **`si`** -- The amount of memory swapped in from disk
- **`so`** -- The amount of memory swapped out to disk

The I/O-related fields are:

- **`bi`** -- Blocks sent to a block device
- **`bo`** -- Blocks received from a block device

The system-related fields are:

- **`in`** -- The number of interrupts per second
- **`cs`** -- The number of context switches per second

The CPU-related fields are:

- **`us`** -- The percentage of the time the CPU ran user-level code
- **`sy`** -- The percentage of the time the CPU ran system-level code
- **`id`** -- The percentage of the time the CPU was idle
- **`wa`** -- I/O wait

When **`vmstat`** is run without any options, only one line is displayed. This line contains averages, calculated from the time the system was last booted.

However, most system administrators do not rely on the data in this line, as the time over which it was collected varies. Instead, most administrators take advantage of **`vmstat`**'s ability to repetitively display resource utilization data at set intervals. For example, the command **`vmstat 1`** displays one new line of utilization data every second, while the command **`vmstat 1 10`** displays one new line per second, but only for the next ten seconds.

In the hands of an experienced administrator, **`vmstat`** can be used to quickly determine resource utilization and performance issues. But to gain more insight into those issues, a different kind of tool is required -- a tool capable of more in-depth data collection and analysis.

2.5.4. The Sysstat Suite of Resource Monitoring Tools

While the previous tools may be helpful for gaining more insight into system performance over very short time frames, they are of little use beyond providing a snapshot of system resource utilization. In addition, there are aspects of system performance that cannot be easily monitored using such simplistic tools.

Therefore, a more sophisticated tool is necessary. Sysstat is such a tool.

Sysstat contains the following tools related to collecting I/O and CPU statistics:

`iostat`
: Displays an overview of CPU utilization, along with I/O statistics for one or more disk drives.

`mpstat`
: Displays more in-depth CPU statistics.

Sysstat also contains tools that collect system resource utilization data and create daily reports based on that data. These tools are:

sadc

Known as the system activity data collector, **sadc** collects system resource utilization information and writes it to a file.

sar

Producing reports from the files created by **sadc**, **sar** reports can be generated interactively or written to a file for more intensive analysis.

The following sections explore each of these tools in more detail.

2.5.4.1. The iostat command

The **iostat** command at its most basic provides an overview of CPU and disk I/O statistics:

```
Linux 2.4.20-1.1931.2.231.2.10.ent (pigdog.example.com) 07/11/2003 avg-cpu: %user %nice %sys
%idle 6.11 2.56 2.15 89.18 Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn dev3-0 1.68
15.69 22.42 31175836 44543290
```

Below the first line (which contains the system's kernel version and hostname, along with the current date), **iostat** displays an overview of the system's average CPU utilization since the last reboot. The CPU utilization report includes the following percentages:

- Percentage of time spent in user mode (running applications, etc.)
- Percentage of time spent in user mode (for processes that have altered their scheduling priority using **nice(2)**)
- Percentage of time spent in kernel mode
- Percentage of time spent idle

Below the CPU utilization report is the device utilization report. This report contains one line for each active disk device on the system and includes the following information:

- The device specification, displayed as **dev<*major-number*>-*sequence-number***, where **<*major-number*>** is the device's major number[3], and **<*sequence-number*>** is a sequence number starting at zero.
- The number of transfers (or I/O operations) per second.
- The number of 512-byte blocks read per second.
- The number of 512-byte blocks written per second.
- The total number of 512-byte blocks read.
- The total number of 512-byte block written.

This is just a sample of the information that can be obtained using **iostat**. For more information, refer to the **iostat(1)** man page.

2.5.4.2. The mpstat command

The **mpstat** command produces the following output:

```
Linux 2.6.11-1.1369_FC4 (example.redhat.com) 02/07/2006 01:22:23 PM CPU %user %nice %system
%iowait %irq %soft %idle intr/s 01:22:23 PM all 0.02 0.00 0.02 0.02 0.02 0.00 99.92 1011.86
```

On multiprocessor systems, **mpstat** allows the utilization for each CPU to be displayed individually, making it possible to determine how effectively each CPU is being used.

2.5.4.3. The sadc command

As stated earlier, the **sadc** command collects system utilization data and writes it to a file for later analysis. By default, the data is written to files in the **/var/log/sa/** directory. The files are named **sa<*dd*>**, where **<*dd*>** is the current day's two-digit date.

sadc is normally run by the **sa1** script. This script is periodically invoked by **cron** via the file **sysstat**, which is located in **/etc/cron.d/**. The **sa1** script invokes **sadc** for a single one-second measuring interval. By default, **cron** runs **sa1** every 10 minutes, adding the data collected during each interval to the current **/var/log/sa/sa<*dd*>** file.

2.5.4.4. The sar command

The **sar** command produces system utilization reports based on the data collected by **sadc**. As configured in Red Hat Enterprise Linux, **sar** is automatically run to process the files automatically collected by **sadc**. The report files are written to **/var/log/sa/** and are named **sar<*dd*>**, where **<*dd*>** is the two-digit representations of the previous day's two-digit date.

sar is normally run by the **sa2** script. This script is periodically invoked by **cron** via the file **sysstat**, which is located in **/etc/cron.d/**. By default, **cron** runs **sa2** once a day at 23:53, allowing it to produce a report for the entire day's data.

2.5.4.4.1. Reading sar Reports

The format of a **sar** report produced by the default Red Hat Enterprise Linux configuration consists of multiple sections, with each section containing a specific type of data, ordered by the time of day that the data was collected. Since **sadc** is configured to perform a one-second measurement interval every ten minutes, the default **sar** reports contain data in ten-minute increments, from 00:00 to 23:50[4].

Each section of the report starts with a heading describing the data contained in the section. The heading is repeated at regular intervals throughout the section, making it easier to interpret the data while paging through the report. Each section ends with a line containing the average of the data reported in that section.

Here is a sample section **sar** report, with the data from 00:30 through 23:40 removed to save space:

```
00:00:01 CPU %user %nice %system %idle 00:10:00 all 6.39 1.96 0.66 90.98 00:20:01 all 1.61
3.16 1.09 94.14 … 23:50:01 all 44.07 0.02 0.77 55.14 Average: all 5.80 4.99 2.87 86.34
```

In this section, CPU utilization information is displayed. This is very similar to the data displayed by **iostat**.

Other sections may have more than one line's worth of data per time, as shown by this section generated from CPU utilization data collected on a dual-processor system:

[4] Due to changing system loads, the actual time at which the data was collected may vary by a second or two.

```
00:00:01 CPU %user %nice %system %idle 00:10:00 0 4.19 1.75 0.70 93.37 00:10:00 1 8.59
2.18 0.63 88.60 00:20:01 0 1.87 3.21 1.14 93.78 00:20:01 1 1.35 3.12 1.04 94.49 … 23:50:01
0 42.84 0.03 0.80 56.33 23:50:01 1 45.29 0.01 0.74 53.95 Average: 0 6.00 5.01 2.74 86.25
Average: 1 5.61 4.97 2.99 86.43
```

There are a total of seventeen different sections present in reports generated by the default Red Hat Enterprise Linux **`sar`** configuration; some are explored in upcoming chapters. For more information about the data contained in each section, refer to the **`sar(1)`** man page.

2.5.5. OProfile

The OProfile system-wide profiler is a low-overhead monitoring tool. OProfile makes use of the processor's performance monitoring hardware[5] to determine the nature of performance-related problems.

Performance monitoring hardware is part of the processor itself. It takes the form of a special counter, incremented each time a certain event (such as the processor not being idle or the requested data not being in cache) occurs. Some processors have more than one such counter and allow the selection of different event types for each counter.

The counters can be loaded with an initial value and produce an interrupt whenever the counter overflows. By loading a counter with different initial values, it is possible to vary the rate at which interrupts are produced. In this way it is possible to control the sample rate and, therefore, the level of detail obtained from the data being collected.

At one extreme, setting the counter so that it generates an overflow interrupt with every event provides extremely detailed performance data (but with massive overhead). At the other extreme, setting the counter so that it generates as few interrupts as possible provides only the most general overview of system performance (with practically no overhead). The secret to effective monitoring is the selection of a sample rate sufficiently high to capture the required data, but not so high as to overload the system with performance monitoring overhead.

Warning

You can configure OProfile so that it produces sufficient overhead to render the system unusable. Therefore, you must exercise care when selecting counter values. For this reason, the **`opcontrol`** command supports the **`--list-events`** option, which displays the event types available for the currently-installed processor, along with suggested minimum counter values for each.

It is important to keep the tradeoff between sample rate and overhead in mind when using OProfile.

2.5.5.1. OProfile Components

Oprofile consists of the following components:

- Data collection software
- Data analysis software
- Administrative interface software

[5] OProfile can also use a fallback mechanism (known as TIMER_INT) for those system architectures that lack performance monitoring hardware.

The data collection software consists of the **`oprofile.o`** kernel module, and the **`oprofiled`** daemon.

The data analysis software includes the following programs:

`op_time`
: Displays the number and relative percentages of samples taken for each executable file

`oprofpp`
: Displays the number and relative percentage of samples taken by either function, individual instruction, or in **`gprof`**-style output

`op_to_source`
: Displays annotated source code and/or assembly listings

`op_visualise`
: Graphically displays collected data

These programs make it possible to display the collected data in a variety of ways.

The administrative interface software controls all aspects of data collection, from specifying which events are to be monitored to starting and stopping the collection itself. This is done using the **`opcontrol`** command.

2.5.5.2. A Sample OProfile Session

This section shows an OProfile monitoring and data analysis session from initial configuration to final data analysis. It is only an introductory overview; for more detailed information, consult the *System Administrators Guide*.

Use **`opcontrol`** to configure the type of data to be collected with the following command:

```
opcontrol \ --vmlinux=/boot/vmlinux-`uname -r` \ --ctr0-event=CPU_CLK_UNHALTED \ --ctr0-
count=6000
```

The options used here direct **`opcontrol`** to:

- Direct OProfile to a copy of the currently running kernel (**``--vmlinux=/boot/vmlinux-`uname -r` ``**)
- Specify that the processor's counter 0 is to be used and that the event to be monitored is the time when the CPU is executing instructions (**`--ctr0-event=CPU_CLK_UNHALTED`**)
- Specify that OProfile is to collect samples every 6000th time the specified event occurs (**`--ctr0-count=6000`**)

Next, check that the **`oprofile`** kernel module is loaded by using the **`lsmod`** command:

```
 Module Size Used by Not tainted oprofile 75616 1 …
```

Confirm that the OProfile file system (located in **`/dev/oprofile/`**) is mounted with the **`ls /dev/oprofile/`** command:

```
 0 buffer buffer_watershed cpu_type enable stats 1 buffer_size cpu_buffer_size dump
 kernel_only
```

(The exact number of files varies according to processor type.)

At this point, the **/root/.oprofile/daemonrc** file contains the settings required by the data collection software:

```
 CTR_EVENT[0]=CPU_CLK_UNHALTED CTR_COUNT[0]=6000 CTR_KERNEL[0]=1 CTR_USER[0]=1 CTR_UM[0]=0
 CTR_EVENT_VAL[0]=121 CTR_EVENT[1]= CTR_COUNT[1]= CTR_KERNEL[1]=1 CTR_USER[1]=1 CTR_UM[1]=0
 CTR_EVENT_VAL[1]= one_enabled=1 SEPARATE_LIB_SAMPLES=0 SEPARATE_KERNEL_SAMPLES=0 VMLINUX=/
boot/vmlinux-2.4.21-1.1931.2.349.2.2.entsmp
```

Next, use **opcontrol** to actually start data collection with the **opcontrol --start** command:

```
 Using log file /var/lib/oprofile/oprofiled.log Daemon started. Profiler running.
```

Verify that the **oprofiled** daemon is running with the command **ps x | grep -i oprofiled**:

```
 32019 ? S 0:00 /usr/bin/oprofiled --separate-lib-samples=0 … 32021 pts/0 S 0:00 grep -i
 oprofiled
```

(The actual **oprofiled** command line displayed by **ps** is much longer; however, it has been truncated here for formatting purposes.)

The system is now being monitored, with the data collected for all executables present on the system. The data is stored in the **/var/lib/oprofile/samples/** directory. The files in this directory follow a somewhat unusual naming convention. Here is an example:

```
 }usr}bin}less#0
```

The naming convention uses the absolute path of each file containing executable code, with the slash (**/**) characters replaced by right curly brackets (**}**), and ending with a pound sign (**#**) followed by a number (in this case, **0**.) Therefore, the file used in this example represents data collected while **/usr/bin/less** was running.

Once data has been collected, use one of the analysis tools to display it. One nice feature of OProfile is that it is not necessary to stop data collection before performing a data analysis. However, you must wait for at least one set of samples to be written to disk, or use the **opcontrol --dump** command to force the samples to disk.

In the following example, **op_time** is used to display (in reverse order -- from highest number of samples to lowest) the samples that have been collected:

```
 3321080 48.8021 0.0000 /boot/vmlinux-2.4.21-1.1931.2.349.2.2.entsmp 761776 11.1940 0.0000 /
usr/bin/oprofiled 368933 5.4213 0.0000 /lib/tls/libc-2.3.2.so 293570 4.3139 0.0000 /usr/lib/
libgobject-2.0.so.0.200.2 205231 3.0158 0.0000 /usr/lib/libgdk-x11-2.0.so.0.200.2 167575
 2.4625 0.0000 /usr/lib/libglib-2.0.so.0.200.2 123095 1.8088 0.0000 /lib/libcrypto.so.0.9.7a
 105677 1.5529 0.0000 /usr/X11R6/bin/XFree86 …
```

Using **less** is a good idea when producing a report interactively, as the reports can be hundreds of lines long. The example given here has been truncated for that reason.

The format for this particular report is that one line is produced for each executable file for which samples were taken. Each line follows this format:

```
<sample-count> <sample-percent> <unused-field> <executable-name>
```

Where:

- ***<sample-count>*** represents the number of samples collected
- ***<sample-percent>*** represents the percentage of all samples collected for this specific executable
- ***<unused-field>*** is a field that is not used
- ***<executable-name>*** represents the name of the file containing executable code for which samples were collected.

This report (produced on a mostly-idle system) shows that nearly half of all samples were taken while the CPU was running code within the kernel itself. Next in line was the OProfile data collection daemon, followed by a variety of libraries and the X Window System server, **XFree86**. It is worth noting that for the system running this sample session, the counter value of 6000 used represents the minimum value recommended by **opcontrol --list-events**. This means that -- at least for this particular system -- OProfile overhead at its highest consumes roughly 11% of the CPU.

2.6. Additional Resources

This section includes various resources that can be used to learn more about resource monitoring and the Red Hat Enterprise Linux-specific subject matter discussed in this chapter.

2.6.1. Installed Documentation

The following resources are installed in the course of a typical Red Hat Enterprise Linux installation.

- **free(1)** man page -- Learn how to display free and used memory statistics.
- **top(1)** man page -- Learn how to display CPU utilization and process-level statistics.
- **watch(1)** man page -- Learn how to periodically execute a user-specified program, displaying fullscreen output.
- **GNOME System MonitorHelp** menu entry -- Learn how to graphically display process, CPU, memory, and disk space utilization statistics.
- **vmstat(8)** man page -- Learn how to display a concise overview of process, memory, swap, I/O, system, and CPU utilization.
- **iostat(1)** man page -- Learn how to display CPU and I/O statistics.
- **mpstat(1)** man page -- Learn how to display individual CPU statistics on multiprocessor systems.
- **sadc(8)** man page -- Learn how to collects system utilization data.
- **sa1(8)** man page -- Learn about a script that runs **sadc** periodically.
- **sar(1)** man page -- Learn how to produce system resource utilization reports.
- **sa2(8)** man page -- Learn how to produce daily system resource utilization report files.

- **`nice(1)`** man page -- Learn how to change process scheduling priority.
- **`oprofile(1)`** man page -- Learn how to profile system performance.
- **`op_visualise(1)`** man page -- Learn how to graphically display OProfile data.

2.6.2. Useful Websites

- *http://people.redhat.com/alikins/system_tuning.html* -- System Tuning Info for Linux Servers. A stream-of-consciousness approach to performance tuning and resource monitoring for servers.
- *http://www.linuxjournal.com/article.php?sid=2396* -- Performance Monitoring Tools for Linux. This Linux Journal page is geared more toward the administrator interested in writing a customized performance graphing solution. Written several years ago, some of the details may no longer apply, but the overall concept and execution are sound.
- *http://oprofile.sourceforge.net/* -- OProfile project website. Includes valuable OProfile resources, including pointers to mailing lists and the #oprofile IRC channel.

2.6.3. Related Books

The following books discuss various issues related to resource monitoring and are good resources for Red Hat Enterprise Linux system administrators:

- The *System Administrators Guide*; Red Hat, Inc -- Includes information on many of the resource monitoring tools described here, including OProfile.
- *Linux Performance Tuning and Capacity Planning* by Jason R. Fink and Matthew D. Sherer; Sams -- Provides more in-depth overviews of the resource monitoring tools presented here and includes others that might be appropriate for more specific resource monitoring needs.
- *Red Hat Linux Security and Optimization* by Mohammed J. Kabir; Red Hat Press -- Approximately the first 150 pages of this book discuss performance-related issues. This includes chapters dedicated to performance issues specific to network, Web, email, and file servers.
- *Linux Administration Handbook* by Evi Nemeth, Garth Snyder, and Trent R. Hein; Prentice Hall -- Provides a short chapter similar in scope to this book, but includes an interesting section on diagnosing a system that has suddenly slowed down.
- *Linux System Administration: A User's Guide* by Marcel Gagne; Addison Wesley Professional -- Contains a small chapter on performance monitoring and tuning.

Bandwidth and Processing Power

Of the two resources discussed in this chapter, one (bandwidth) is often hard for the new system administrator to understand, while the other (processing power) is usually a much easier concept to grasp.

Additionally, it may seem that these two resources are not that closely related -- why group them together?

The reason for addressing both resources together is that these resources are based on the hardware that tie directly into a computer's ability to move and process data. As such, their relationship is often interrelated.

3.1. Bandwidth

At its most basic, bandwidth is the capacity for data transfer -- in other words, how much data can be moved from one point to another in a given amount of time. Having point-to-point data communication implies two things:

- A set of electrical conductors used to make low-level communication possible
- A protocol to facilitate the efficient and reliable communication of data

There are two types of system components that meet these requirements:

- Buses
- Datapaths

The following sections explore each in more detail.

3.1.1. Buses

As stated above, buses enable point-to-point communication and use some sort of protocol to ensure that all communication takes place in a controlled manner. However, buses have other distinguishing features:

- Standardized electrical characteristics (such as the number of conductors, voltage levels, signaling speeds, etc.)
- Standardized mechanical characteristics (such as the type of connector, card size, physical layout, etc.)
- Standardized protocol

The word "standardized" is important because buses are the primary way in which different system components are connected together.

In many cases, buses allow the interconnection of hardware made by multiple manufacturers; without standardization, this would not be possible. However, even in situations where a bus is proprietary to one manufacturer, standardization is important because it allows that manufacturer to more easily implement different components by using a common interface -- the bus itself.

3.1.1.1. Examples of Buses

No matter where in a computer system you look, there are buses. Here are a few of the more common ones:

- Mass storage buses (ATA and SCSI)
- Networks[1] (Ethernet and Token Ring)
- Memory buses (PC133 and Rambus®)
- Expansion buses (PCI, ISA, USB)

3.1.2. Datapaths

Datapaths can be harder to identify but, like buses, they are everywhere. Also like buses, datapaths enable point-to-point communication. However, unlike buses, datapaths:

- Use a simpler protocol (if any)
- Have little (if any) mechanical standardization

The reason for these differences is that datapaths are normally internal to some system component and are not used to facilitate the ad-hoc interconnection of different components. As such, datapaths are highly optimized for a particular situation, where speed and low cost are preferred over slower and more expensive general-purpose flexibility.

3.1.2.1. Examples of Datapaths

Here are some typical datapaths:

- CPU to on-chip cache datapath
- Graphics processor to video memory datapath

3.1.3. Potential Bandwidth-Related Problems

There are two ways in which bandwidth-related problems may occur (for either buses or datapaths):

1. The bus or datapath may represent a shared resource. In this situation, high levels of contention for the bus reduces the effective bandwidth available for all devices on the bus.

   A SCSI bus with several highly-active disk drives would be a good example of this. The highly-active disk drives saturate the SCSI bus, leaving little bandwidth available for any other device on the same bus. The end result is that all I/O to any of the devices on this bus is slow, even if each device on the bus is not overly active.

2. The bus or datapath may be a dedicated resource with a fixed number of devices attached to it. In this case, the electrical characteristics of the bus (and to some extent the nature of the protocol being used) limit the available bandwidth. This is usually more the case with datapaths than with buses. This is one reason why graphics adapters tend to perform more slowly when operating at higher resolutions and/or color depths -- for every screen refresh, there is more data that must be passed along the datapath connecting video memory and the graphics processor.

3.1.4. Potential Bandwidth-Related Solutions

Fortunately, bandwidth-related problems can be addressed. In fact, there are several approaches you can take:

- Spread the load
- Reduce the load

- Increase the capacity

The following sections explore each approach in more detail.

3.1.4.1. Spread the Load

The first approach is to more evenly distribute the bus activity. In other words, if one bus is overloaded and another is idle, perhaps the situation would be improved by moving some of the load to the idle bus.

As a system administrator, this is the first approach you should consider, as often there are additional buses already present in your system. For example, most PCs include at least two ATA *channels* (which is just another name for a bus). If you have two ATA disk drives and two ATA channels, why should both drives be on the same channel?

Even if your system configuration does not include additional buses, spreading the load might still be a reasonable approach. The hardware expenditures to do so would be less expensive than replacing an existing bus with higher-capacity hardware.

3.1.4.2. Reduce the Load

At first glance, reducing the load and spreading the load appear to be different sides of the same coin. After all, when one spreads the load, it acts to reduce the load (at least on the overloaded bus), correct?

While this viewpoint is correct, it is not the same as reducing the load *globally*. The key here is to determine if there is some aspect of the system load that is causing this particular bus to be overloaded. For example, is a network heavily loaded due to activities that are unnecessary? Perhaps a small temporary file is the recipient of heavy read/write I/O. If that temporary file resides on a networked file server, a great deal of network traffic could be eliminated by working with the file locally.

3.1.4.3. Increase the Capacity

The obvious solution to insufficient bandwidth is to increase it somehow. However, this is usually an expensive proposition. Consider, for example, a SCSI controller and its overloaded bus. To increase its bandwidth, the SCSI controller (and likely all devices attached to it) would need to be replaced with faster hardware. If the SCSI controller is a separate card, this would be a relatively straightforward process, but if the SCSI controller is part of the system's motherboard, it becomes much more difficult to justify the economics of such a change.

3.1.5. In Summary…

All system administrators should be aware of bandwidth, and how system configuration and usage impacts available bandwidth. Unfortunately, it is not always apparent what is a bandwidth-related problem and what is not. Sometimes, the problem is not the bus itself, but one of the components attached to the bus.

For example, consider a SCSI adapter that is connected to a PCI bus. If there are performance problems with SCSI disk I/O, it might be the result of a poorly-performing SCSI adapter, even though the SCSI and PCI buses themselves are nowhere near their bandwidth capabilities.

3.2. Processing Power

Often known as CPU power, CPU cycles, and various other names, processing power is the ability of a computer to manipulate data. Processing power varies with the architecture (and clock speed) of the CPU -- usually CPUs with higher clock speeds and those supporting larger word sizes have more processing power than slower CPUs supporting smaller word sizes.

3.2.1. Facts About Processing Power

Here are the two main facts about processing power that you should keep in mind:

- Processing power is fixed
- Processing power cannot be stored

Processing power is fixed, in that the CPU can only go so fast. For example, if you need to add two numbers together (an operation that takes only one machine instruction on most architectures), a particular CPU can do it at one speed, and one speed only. With few exceptions, it is not even possible to *slow* the rate at which a CPU processes instructions, much less increase it.

Processing power is also fixed in another way: it is finite. That is, there are limits to the types of CPUs that can be plugged into any given computer. Some systems are capable of supporting a wide range of CPUs of differing speeds, while others may not be upgradeable at all[2].

Processing power cannot be stored for later use. In other words, if a CPU can process 100 million instructions in one second, one second of idle time equals 100 million instructions worth of processing that have been wasted.

If we take these facts and examine them from a slightly different perspective, a CPU "produces" a stream of executed instructions at a fixed rate. And if the CPU "produces" executed instructions, that means that something else must "consume" them. The next section defines these consumers.

3.2.2. Consumers of Processing Power

There are two main consumers of processing power:

- Applications
- The operating system itself

3.2.2.1. Applications

The most obvious consumers of processing power are the applications and programs you want the computer to run for you. From a spreadsheet to a database, applications are the reason you have a computer.

A single-CPU system can only do one thing at any given time. Therefore, if your application is running, everything else on the system is not. And the opposite is, of course, true -- if something other than your application is running, then your application is doing nothing.

But how is it that many different applications can seemingly run at once under a modern operating system? The answer is that these are multitasking operating systems. In other words, they create the illusion that many different things are going on simultaneously when in fact that is not possible. The trick is to give each process a fraction of a second's worth of time running on the CPU before giving the CPU to another process for the next fraction of a second. If these *context switches* happen frequently enough, the illusion of multiple applications running simultaneously is achieved.

[2] This situation leads to what is humorously termed as a *forklift upgrade*, which means a complete replacement of a computer.

Of course, applications do other things than manipulate data using the CPU. They may wait for user input as well as performing I/O to devices such as disk drives and graphics displays. When these events take place, the application no longer needs the CPU. At these times, the CPU can be used for other processes running other applications without slowing the waiting application at all.

In addition, the CPU can be used by another consumer of processing power: the operating system itself.

3.2.2.2. The Operating System

It is difficult to determine how much processing power is consumed by the operating system. The reason for this is that operating systems use a mixture of process-level and system-level code to perform their work. While, for example, it is easy to use a process monitor to determine what the process running a *daemon* or *service* is doing, it is not so easy to determine how much processing power is being consumed by system-level I/O-related processing (which is normally done within the context of the process requesting the I/O.)

In general, it is possible to divide this kind of operating system overhead into two types:

- Operating system housekeeping
- Process-related activities

Operating system housekeeping includes activities such as process scheduling and memory management, while process-related activities include any processes that support the operating system itself, such as processes handling system-wide event logging or I/O cache flushing.

3.2.3. Improving a CPU Shortage

When there is insufficient processing power available for the work needing to be done, you have two options:

- Reducing the load
- Increasing the capacity

3.2.3.1. Reducing the Load

Reducing the CPU load is something that can be done with no expenditure of money. The trick is to identify those aspects of the system load under your control that can be cut back. There are three areas to focus on:

- Reducing operating system overhead
- Reducing application overhead
- Eliminating applications entirely

3.2.3.1.1. Reducing Operating System Overhead

To reduce operating system overhead, you must examine your current system load and determine what aspects of it result in inordinate amounts of overhead. These areas could include:

- Reducing the need for frequent process scheduling
- Reducing the amount of I/O performed

Do not expect miracles; in a reasonably-well configured system, it is unlikely to notice much of a performance increase by trying to reduce operating system overhead. This is due to the fact that a reasonably-well configured system, by definition, results in a minimal amount of overhead. However, if your system is running with too little RAM for instance, you may be able to reduce overhead by alleviating the RAM shortage.

3.2.3.1.2. Reducing Application Overhead

Reducing application overhead means making sure the application has everything it needs to run well. Some applications exhibit wildly different behaviors under different environments -- an application may become highly compute-bound while processing certain types of data, but not others, for example.

The point to keep in mind here is that you must understand the applications running on your system if you are to enable them to run as efficiently as possible. Often this entails working with your users, and/or your organization's developers, to help uncover ways in which the applications can be made to run more efficiently.

3.2.3.1.3. Eliminating Applications Entirely

Depending on your organization, this approach might not be available to you, as it often is not a system administrator's responsibility to dictate which applications will and will not be run. However, if you can identify any applications that are known "CPU hogs", you might be able to influence the powers-that-be to retire them.

Doing this will likely involve more than just yourself. The affected users should certainly be a part of this process; in many cases they may have the knowledge and the political power to make the necessary changes to the application lineup.

Note

Keep in mind that an application may not need to be eliminated from every system in your organization. You might be able to move a particularly CPU-hungry application from an overloaded system to another system that is nearly idle.

3.2.3.2. Increasing the Capacity

Of course, if it is not possible to reduce the demand for processing power, you must find ways of increasing the processing power that is available. To do so costs money, but it can be done.

3.2.3.2.1. Upgrading the CPU

The most straightforward approach is to determine if your system's CPU can be upgraded. The first step is to determine if the current CPU can be removed. Some systems (primarily laptops) have CPUs that are soldered in place, making an upgrade impossible. The rest, however, have socketed CPUs, making upgrades possible -- at least in theory.

Next, you must do some research to determine if a faster CPU exists for your system configuration. For example, if you currently have a 1GHz CPU, and a 2GHz unit of the same type exists, an upgrade might be possible.

Finally, you must determine the maximum clock speed supported by your system. To continue the example above, even if a 2GHz CPU of the proper type exists, a simple CPU swap is not an option if your system only supports processors running at 1GHz or below.

Should you find that you cannot install a faster CPU in your system, your options may be limited to changing motherboards or even the more expensive forklift upgrade mentioned earlier.

However, some system configurations make a slightly different approach possible. Instead of replacing the current CPU, why not just add another one?

3.2.3.2.2. Is Symmetric Multiprocessing Right for You?

Symmetric multiprocessing (also known as SMP) makes it possible for a computer system to have more than one CPU sharing all system resources. This means that, unlike a uniprocessor system, an SMP system may actually have more than one process running at the same time.

At first glance, this seems like any system administrator's dream. First and foremost, SMP makes it possible to increase a system's CPU power even if CPUs with faster clock speeds are not available -- just by adding another CPU. However, this flexibility comes with some caveats.

The first caveat is that not all systems are capable of SMP operation. Your system must have a motherboard designed to support multiple processors. If it does not, a motherboard upgrade (at the least) would be required.

The second caveat is that SMP increases system overhead. This makes sense if you stop to think about it; with more CPUs to schedule work for, the operating system requires more CPU cycles for overhead. Another aspect to this is that with multiple CPUs, there can be more contention for system resources. Because of these factors, upgrading a dual-processor system to a quad-processor unit does not result in a 100% increase in available CPU power. In fact, depending on the actual hardware, the workload, and the processor architecture, it is possible to reach a point where the addition of another processor could actually *reduce* system performance.

Another point to keep in mind is that SMP does not help workloads consisting of one monolithic application with a single stream of execution. In other words, if a large compute-bound simulation program runs as one process and without threads, it will not run any faster on an SMP system than on a single-processor machine. In fact, it may even run somewhat slower, due to the increased overhead SMP brings. For these reasons, many system administrators feel that when it comes to CPU power, single stream processing power is the way to go. It provides the most CPU power with the fewest restrictions on its use.

While this discussion seems to indicate that SMP is never a good idea, there are circumstances in which it makes sense. For example, environments running multiple highly compute-bound applications are good candidates for SMP. The reason for this is that applications that do nothing but compute for long periods of time keep contention between active processes (and therefore, the operating system overhead) to a minimum, while the processes themselves keep every CPU busy.

One other thing to keep in mind about SMP is that the performance of an SMP system tends to degrade more gracefully as the system load increases. This does make SMP systems popular in server and multi-user environments, as the ever-changing process mix can impact the system-wide load less on a multi-processor machine.

3.3. Red Hat Enterprise Linux-Specific Information

Monitoring bandwidth and CPU utilization under Red Hat Enterprise Linux entails using the tools discussed in *Chapter 2, Resource Monitoring*; therefore, if you have not yet read that chapter, you should do so before continuing.

3.3.1. Monitoring Bandwidth on Red Hat Enterprise Linux

As stated in *Section 2.4.2, “Monitoring Bandwidth”*, it is difficult to directly monitor bandwidth utilization. However, by examining device-level statistics, it is possible to roughly gauge whether insufficient bandwidth is an issue on your system.

By using **vmstat**, it is possible to determine if overall device activity is excessive by examining the **bi** and **bo** fields; in addition, taking note of the **si** and **so** fields give you a bit more insight into how much disk activity is due to swap-related I/O:

```
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu---- r b swpd free
buff cache si so bi bo in cs us sy id wa 1 0 0 248088 158636 480804 0 0 2 6 120 120 10 3 87
0
```

In this example, the **bi** field shows two blocks/second written to block devices (primarily disk drives), while the **bo** field shows six blocks/second read from block devices. We can determine that none of this activity was due to swapping, as the **si** and **so** fields both show a swap-related I/O rate of zero kilobytes/second.

By using **iostat**, it is possible to gain a bit more insight into disk-related activity:

```
Linux 2.4.21-1.1931.2.349.2.2.entsmp (raptor.example.com) 07/21/2003 avg-cpu: %user %nice
%sys %idle 5.34 4.60 2.83 87.24 Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn dev8-0
1.10 6.21 25.08 961342 3881610 dev8-1 0.00 0.00 0.00 16 0
```

This output shows us that the device with major number 8 (which is **/dev/sda**, the first SCSI disk) averaged slightly more than one I/O operation per second (the **tsp** field). Most of the I/O activity for this device were writes (the **Blk_wrtn** field), with slightly more than 25 blocks written each second (the **Blk_wrtn/s** field).

If more detail is required, use **iostat**'s **-x** option:

```
Linux 2.4.21-1.1931.2.349.2.2.entsmp (raptor.example.com) 07/21/2003 avg-cpu: %user %nice
%sys %idle 5.37 4.54 2.81 87.27 Device: rrqm/s wrqm/s r/s w/s rsec/s wsec/s rkB/s wkB/s
avgrq-sz /dev/sda 13.57 2.86 0.36 0.77 32.20 29.05 16.10 14.53 54.52 /dev/sda1 0.17 0.00
0.00 0.00 0.34 0.00 0.17 0.00 133.40 /dev/sda2 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
11.56 /dev/sda3 0.31 2.11 0.29 0.62 4.74 21.80 2.37 10.90 29.42 /dev/sda4 0.09 0.75 0.04
0.15 1.06 7.24 0.53 3.62 43.01
```

Over and above the longer lines containing more fields, the first thing to keep in mind is that this **iostat** output is now displaying statistics on a per-partition level. By using **df** to associate mount points with device names, it is possible to use this report to determine if, for example, the partition containing **/home/** is experiencing an excessive workload.

Actually, each line output from **iostat -x** is longer and contains more information than this; here is the remainder of each line (with the device column added for easier reading):

```
Device: avgqu-sz await svctm %util /dev/sda 0.24 20.86 3.80 0.43 /dev/sda1 0.00 141.18
122.73 0.03 /dev/sda2 0.00 6.00 6.00 0.00 /dev/sda3 0.12 12.84 2.68 0.24 /dev/sda4 0.11
57.47 8.94 0.17
```

In this example, it is interesting to note that **/dev/sda2** is the system swap partition; it is obvious from the many fields reading **0.00** for this partition that swapping is not a problem on this system.

Another interesting point to note is **`/dev/sda1`**. The statistics for this partition are unusual; the overall activity seems low, but why are the average I/O request size (the **`avgrq-sz`** field), average wait time (the **`await`** field), and the average service time (the **`svctm`** field) so much larger than the other partitions? The answer is that this partition contains the **`/boot/`** directory, which is where the kernel and initial ramdisk are stored. When the system boots, the read I/Os (notice that only the **`rsec/s`** and **`rkB/s`** fields are non-zero; no writing is done here on a regular basis) used during the boot process are for large numbers of blocks, resulting in the relatively long wait and service times **`iostat`** displays.

It is possible to use **`sar`** for a longer-term overview of I/O statistics; for example, **`sar -b`** displays a general I/O report:

```
Linux 2.4.21-1.1931.2.349.2.2.entsmp (raptor.example.com) 07/21/2003 12:00:00 AM tps rtps
wtps bread/s bwrtn/s 12:10:00 AM 0.51 0.01 0.50 0.25 14.32 12:20:01 AM 0.48 0.00 0.48 0.00
13.32 … 06:00:02 PM 1.24 0.00 1.24 0.01 36.23 Average: 1.11 0.31 0.80 68.14 34.79
```

Here, like **`iostat`**'s initial display, the statistics are grouped for all block devices.

Another I/O-related report is produced using **`sar -d`**:

```
Linux 2.4.21-1.1931.2.349.2.2.entsmp (raptor.example.com) 07/21/2003 12:00:00 AM DEV tps
sect/s 12:10:00 AM dev8-0 0.51 14.57 12:10:00 AM dev8-1 0.00 0.00 12:20:01 AM dev8-0 0.48
13.32 12:20:01 AM dev8-1 0.00 0.00 … 06:00:02 PM dev8-0 1.24 36.25 06:00:02 PM dev8-1 0.00
0.00 Average: dev8-0 1.11 102.93 Average: dev8-1 0.00 0.00
```

This report provides per-device information, but with little detail.

While there are no explicit statistics showing bandwidth utilization for a given bus or datapath, we can at least determine what the devices are doing and use their activity to indirectly determine the bus loading.

3.3.2. Monitoring CPU Utilization on Red Hat Enterprise Linux

Unlike bandwidth, monitoring CPU utilization is much more straightforward. From a single percentage of CPU utilization in **GNOME System Monitor**, to the more in-depth statistics reported by **`sar`**, it is possible to accurately determine how much CPU power is being consumed and by what.

Moving beyond **GNOME System Monitor**, **`top`** is the first resource monitoring tool discussed in *Chapter 2, Resource Monitoring* to provide a more in-depth representation of CPU utilization. Here is a **`top`** report from a dual-processor workstation:

```
9:44pm up 2 days, 2 min, 1 user, load average: 0.14, 0.12, 0.09 90 processes: 82 sleeping,
1 running, 7 zombie, 0 stopped CPU0 states: 0.4% user, 1.1% system, 0.0% nice, 97.4% idle
CPU1 states: 0.5% user, 1.3% system, 0.0% nice, 97.1% idle Mem: 1288720K av, 1056260K used,
232460K free, 0K shrd, 145644K buff Swap: 522104K av, 0K used, 522104K free 469764K cached
PID USER PRI NI SIZE RSS SHARE STAT %CPU %MEM TIME COMMAND 30997 ed 16 0 1100 1100 840 R
1.7 0.0 0:00 top 1120 root 5 -10 249M 174M 71508 S 0.9 13.8 254:59 X 1260 ed 15 0 54408 53M
6864 S 0.7 4.2 12:09 gnome-terminal 888 root 15 0 2428 2428 1796 S 0.1 0.1 0:06 sendmail
1264 ed 15 0 16336 15M 9480 S 0.1 1.2 1:58 rhn-applet-gui 1 root 15 0 476 476 424 S 0.0 0.0
0:05 init 2 root 0K 0 0 0 0 SW 0.0 0.0 0:00 migration_CPU0 3 root 0K 0 0 0 0 SW 0.0 0.0 0:00
migration_CPU1 4 root 15 0 0 0 0 SW 0.0 0.0 0:01 keventd 5 root 34 19 0 0 0 SWN 0.0 0.0 0:00
ksoftirqd_CPU0 6 root 34 19 0 0 0 SWN 0.0 0.0 0:00 ksoftirqd_CPU1 7 root 15 0 0 0 0 SW 0.0
0.0 0:05 kswapd 8 root 15 0 0 0 0 SW 0.0 0.0 0:00 bdflush 9 root 15 0 0 0 0 SW 0.0 0.0 0:01
kupdated 10 root 25 0 0 0 0 SW 0.0 0.0 0:00 mdrecoveryd
```

The first CPU-related information is present on the very first line: the load average. The load average is a number corresponding to the average number of runnable processes on the system. The load average is often listed as three sets of numbers (as **top** does), which represent the load average for the past 1, 5, and 15 minutes, indicating that the system in this example was not very busy.

The next line, although not strictly related to CPU utilization, has an indirect relationship, in that it shows the number of runnable processes (here, only one -- remember this number, as it means something special in this example). The number of runnable processes is a good indicator of how CPU-bound a system might be.

Next are two lines displaying the current utilization for each of the two CPUs in the system. The utilization statistics show whether the CPU cycles were expended for user-level or system-level processing; also included is a statistic showing how much CPU time was expended by processes with altered scheduling priorities. Finally, there is an idle time statistic.

Moving down into the process-related section of the display, we find that the process using the most CPU power is **top** itself; in other words, the one runnable process on this otherwise-idle system was **top** taking a "picture" of itself.

Note

It is important to remember that the very act of running a system monitor affects the resource utilization statistics you receive. All software-based monitors do this to some extent.

To gain more detailed knowledge regarding CPU utilization, we must change tools. If we examine output from **vmstat**, we obtain a slightly different understanding of our example system:

```
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu---- r b swpd free
buff cache si so bi bo in cs us sy id wa 1 0 0 233276 146636 469808 0 0 7 7 14 27 10 3 87 0
0 0 0 233276 146636 469808 0 0 0 0 523 138 3 0 96 0 0 0 0 233276 146636 469808 0 0 0 0 557
385 2 1 97 0 0 0 0 233276 146636 469808 0 0 0 0 544 343 2 0 97 0 0 0 0 233276 146636 469808
0 0 0 0 517 89 2 0 98 0 0 0 0 233276 146636 469808 0 0 0 32 518 102 2 0 98 0 0 0 0 233276
146636 469808 0 0 0 0 516 91 2 1 98 0 0 0 0 233276 146636 469808 0 0 0 0 516 72 2 0 98 0 0 0
0 233276 146636 469808 0 0 0 0 516 88 2 0 97 0 0 0 0 233276 146636 469808 0 0 0 0 516 81 2 0
97 0
```

Here we have used the command **vmstat 1 10** to sample the system every second for ten times. At first, the CPU-related statistics (the **us**, **sy**, and **id** fields) seem similar to what **top** displayed, and maybe even appear a bit less detailed. However, unlike **top**, we can also gain a bit of insight into how the CPU is being used.

If we examine at the **system** fields, we notice that the CPU is handling about 500 interrupts per second on average and is switching between processes anywhere from 80 to nearly 400 times a second. If you think this seems like a lot of activity, think again, because the user-level processing (the **us** field) is only averaging 2%, while system-level processing (the **sy** field) is usually under 1%. Again, this is an idle system.

Reviewing the tools Sysstat offers, we find that **iostat** and **mpstat** provide little additional information over what we have already experienced with **top** and **vmstat**. However, **sar** produces a number of reports that can come in handy when monitoring CPU utilization.

The first report is obtained by the command **sar -q**, which displays the run queue length, total number of processes, and the load averages for the past one and five minutes. Here is a sample:

```
Linux 2.4.21-1.1931.2.349.2.2.entsmp (falcon.example.com) 07/21/2003 12:00:01 AM runq-sz
plist-sz ldavg-1 ldavg-5 12:10:00 AM 3 122 0.07 0.28 12:20:01 AM 5 123 0.00 0.03 … 09:50:00
AM 5 124 0.67 0.65 Average: 4 123 0.26 0.26
```

In this example, the system is always busy (given that more than one process is runnable at any given time), but is not overly loaded (due to the fact that this particular system has more than one processor).

The next CPU-related **`sar`** report is produced by the command **`sar -u`**:

```
Linux 2.4.21-1.1931.2.349.2.2.entsmp (falcon.example.com) 07/21/2003 12:00:01 AM CPU %user
%nice %system %idle 12:10:00 AM all 3.69 20.10 1.06 75.15 12:20:01 AM all 1.73 0.22 0.80
97.25 … 10:00:00 AM all 35.17 0.83 1.06 62.93 Average: all 7.47 4.85 3.87 83.81
```

The statistics contained in this report are no different from those produced by many of the other tools. The biggest benefit here is that **`sar`** makes the data available on an ongoing basis and is therefore more useful for obtaining long-term averages, or for the production of CPU utilization graphs.

On multiprocessor systems, the **`sar -U`** command can produce statistics for an individual processor or for all processors. Here is an example of output from **`sar -U ALL`**:

```
Linux 2.4.21-1.1931.2.349.2.2.entsmp (falcon.example.com) 07/21/2003 12:00:01 AM CPU %user
%nice %system %idle 12:10:00 AM 0 3.46 21.47 1.09 73.98 12:10:00 AM 1 3.91 18.73 1.03 76.33
12:20:01 AM 0 1.63 0.25 0.78 97.34 12:20:01 AM 1 1.82 0.20 0.81 97.17 … 10:00:00 AM 0 39.12
0.75 1.04 59.09 10:00:00 AM 1 31.22 0.92 1.09 66.77 Average: 0 7.61 4.91 3.86 83.61 Average:
1 7.33 4.78 3.88 84.02
```

The **`sar -w`** command reports on the number of context switches per second, making it possible to gain additional insight in where CPU cycles are being spent:

```
Linux 2.4.21-1.1931.2.349.2.2.entsmp (falcon.example.com) 07/21/2003 12:00:01 AM cswch/s
12:10:00 AM 537.97 12:20:01 AM 339.43 … 10:10:00 AM 319.42 Average: 1158.25
```

It is also possible to produce two different **`sar`** reports on interrupt activity. The first, (produced using the **`sar -I SUM`** command) displays a single "interrupts per second" statistic:

```
Linux 2.4.21-1.1931.2.349.2.2.entsmp (falcon.example.com) 07/21/2003 12:00:01 AM INTR intr/s
12:10:00 AM sum 539.15 12:20:01 AM sum 539.49 … 10:40:01 AM sum 539.10 Average: sum 541.00
```

By using the command **`sar -I PROC`**, it is possible to break down interrupt activity by processor (on multiprocessor systems) *and* by interrupt level (from 0 to 15):

```
 Linux 2.4.21-1.1931.2.349.2.2.entsmp (pigdog.example.com) 07/21/2003 12:00:00 AM CPU i000/s
 i001/s i002/s i008/s i009/s i011/s i012/s 12:10:01 AM 0 512.01 0.00 0.00 0.00 3.44 0.00 0.00
 12:10:01 AM CPU i000/s i001/s i002/s i008/s i009/s i011/s i012/s 12:20:01 AM 0 512.00 0.00
 0.00 0.00 3.73 0.00 0.00 … 10:30:01 AM CPU i000/s i001/s i002/s i003/s i008/s i009/s i010/
s 10:40:02 AM 0 512.00 1.67 0.00 0.00 0.00 15.08 0.00 Average: 0 512.00 0.42 0.00 N/A 0.00
 6.03 N/A
```

This report (which has been truncated horizontally to fit on the page) includes one column for each interrupt level (for example, the **`i002/s`** field illustrating the rate for interrupt level 2). If this were a multiprocessor system, there would be one line per sample period for each CPU.

Another important point to note about this report is that **sar** adds or removes specific interrupt fields if no data is collected for that field. The example report above provides an example of this, the end of the report includes interrupt levels (3 and 10) that were not present at the start of the sampling period.

Note

There are two other interrupt-related **sar** reports -- **sar -I ALL** and **sar -I XALL**. However, the default configuration for the **sadc** data collection utility does not collect the information necessary for these reports. This can be changed by editing the file **/etc/cron.d/sysstat**, and changing this line:

```
*/10 * * * * root /usr/lib/sa/sa1 1 1
```

to this:

```
*/10 * * * * root /usr/lib/sa/sa1 -I 1 1
```

Keep in mind this change does cause additional information to be collected by **sadc**, and results in larger data file sizes. Therefore, make sure your system configuration can support the additional space consumption.

3.4. Additional Resources

This section includes various resources that can be used to learn more about the Red Hat Enterprise Linux-specific subject matter discussed in this chapter.

3.4.1. Installed Documentation

The following resources are installed in the course of a typical Red Hat Enterprise Linux installation and can help you learn more about the subject matter discussed in this chapter.

- **vmstat(8)** man page -- Learn how to display a concise overview of process, memory, swap, I/O, system, and CPU utilization.
- **iostat(1)** man page -- Learn how to display CPU and I/O statistics.
- **sar(1)** man page -- Learn how to produce system resource utilization reports.
- **sadc(8)** man page -- Learn how to collect system utilization data.
- **sa1(8)** man page -- Learn about a script that runs **sadc** periodically.
- **top(1)** man page -- Learn how to display CPU utilization and process-level statistics.

3.4.2. Useful Websites

- *http://people.redhat.com/alikins/system_tuning.html* -- System Tuning Info for Linux Servers. A stream-of-consciousness approach to performance tuning and resource monitoring for servers.
- *http://www.linuxjournal.com/article.php?sid=2396* -- Performance Monitoring Tools for Linux. This Linux Journal page is geared more toward the administrator interested in writing a customized performance graphing solution. Written several years ago, some of the details may no longer apply, but the overall concept and execution are sound.

3.4.3. Related Books

The following books discuss various issues related to resource monitoring, and are good resources for Red Hat Enterprise Linux system administrators:

- The *System Administrators Guide*; Red Hat, Inc -- Includes a chapter on many of the resource monitoring tools described here.
- *Linux Performance Tuning and Capacity Planning* by Jason R. Fink and Matthew D. Sherer; Sams -- Provides more in-depth overviews of the resource monitoring tools presented here, and includes others that might be appropriate for more specific resource monitoring needs.
- *Linux Administration Handbook* by Evi Nemeth, Garth Snyder, and Trent R. Hein; Prentice Hall -- Provides a short chapter similar in scope to this book, but includes an interesting section on diagnosing a system that has suddenly slowed down.
- *Linux System Administration: A User's Guide* by Marcel Gagne; Addison Wesley Professional -- Contains a small chapter on performance monitoring and tuning.

Physical and Virtual Memory

All present-day, general-purpose computers are of the type known as *stored program computers*. As the name implies, stored program computers load instructions (the building blocks of programs) into some type of internal storage, where they subsequently execute those instructions.

Stored program computers also use the same storage for data. This is in contrast to computers that use their hardware configuration to control their operation (such as older plugboard-based computers).

The place where programs were stored on the first stored program computers went by a variety of names and used a variety of different technologies, from spots on a cathode ray tube, to pressure pulses in columns of mercury. Fortunately, present-day computers use technologies with greater storage capacity and much smaller size than ever before.

4.1. Storage Access Patterns

One thing to keep in mind throughout this chapter is that computers tend to access storage in certain ways. In fact, most storage access tends to exhibit one (or both) of the following attributes:

- Access tends to be sequential
- Access tends to be localized

Sequential access means that, if address *N* is accessed by the CPU, it is highly likely that address *N*+1 will be accessed next. This makes sense, as most programs consist of large sections of instructions that execute -- in order -- one after the other.

Localized access means that, if address *X* is accessed, it is likely that other addresses surrounding *X* will also be accessed in the future.

These attributes are crucial, because it allows smaller, faster storage to effectively buffer larger, slower storage. This is the basis for implementing virtual memory. But before we can discuss virtual memory, we must examine the various storage technologies currently in use.

4.2. The Storage Spectrum

Present-day computers actually use a variety of storage technologies. Each technology is geared toward a specific function, with speeds and capacities to match.

These technologies are:

- CPU registers
- Cache memory
- RAM
- Hard drives
- Off-line backup storage (tape, optical disk, etc.)

In terms of capabilities and cost, these technologies form a spectrum. For example, CPU registers are:

- Very fast (access times of a few nanoseconds)
- Low capacity (usually less than 200 bytes)
- Very limited expansion capabilities (a change in CPU architecture would be required)

- Expensive (more than one dollar/byte)

However, at the other end of the spectrum, off-line backup storage is:

- Very slow (access times may be measured in days, if the backup media must be shipped long distances)
- Very high capacity (10s - 100s of gigabytes)
- Essentially unlimited expansion capabilities (limited only by the floorspace needed to house the backup media)
- Very inexpensive (fractional cents/byte)

By using different technologies with different capabilities, it is possible to fine-tune system design for maximum performance at the lowest possible cost. The following sections explore each technology in the storage spectrum.

4.2.1. CPU Registers

Every present-day CPU design includes registers for a variety of purposes, from storing the address of the currently-executed instruction to more general-purpose data storage and manipulation. CPU registers run at the same speed as the rest of the CPU; otherwise, they would be a serious bottleneck to overall system performance. The reason for this is that nearly all operations performed by the CPU involve the registers in one way or another.

The number of CPU registers (and their uses) are strictly dependent on the architectural design of the CPU itself. There is no way to change the number of CPU registers, short of migrating to a CPU with a different architecture. For these reasons, the number of CPU registers can be considered a constant, as they are changeable only with great pain and expense.

4.2.2. Cache Memory

The purpose of cache memory is to act as a buffer between the very limited, very high-speed CPU registers and the relatively slower and much larger main system memory -- usually referred to as RAM[1]. Cache memory has an operating speed similar to the CPU itself so, when the CPU accesses data in cache, the CPU is not kept waiting for the data.

Cache memory is configured such that, whenever data is to be read from RAM, the system hardware first checks to determine if the desired data is in cache. If the data is in cache, it is quickly retrieved, and used by the CPU. However, if the data is not in cache, the data is read from RAM and, while being transferred to the CPU, is also placed in cache (in case it is needed again later). From the perspective of the CPU, all this is done transparently, so that the only difference between accessing data in cache and accessing data in RAM is the amount of time it takes for the data to be returned.

In terms of storage capacity, cache is much smaller than RAM. Therefore, not every byte in RAM can have its own unique location in cache. As such, it is necessary to split cache up into sections that can be used to cache different areas of RAM, and to have a mechanism that allows each area of cache to cache different areas of RAM at different times. Even with the difference in size between cache and RAM, given the sequential and localized nature of storage access, a small amount of cache can effectively speed access to a large amount of RAM.

[1] While "RAM" is an acronym for "Random Access Memory," and a term that could easily apply to any storage technology allowing the non-sequential access of stored data, when system administrators talk about RAM they invariably mean main system memory.

When writing data from the CPU, things get a bit more complicated. There are two different approaches that can be used. In both cases, the data is first written to cache. However, since the purpose of cache is to function as a very fast copy of the contents of selected portions of RAM, any time a piece of data changes its value, that new value must be written to both cache memory and RAM. Otherwise, the data in cache and the data in RAM would no longer match.

The two approaches differ in how this is done. One approach, known as *write-through* caching, immediately writes the modified data to RAM. *Write-back* caching, however, delays the writing of modified data back to RAM. The reason for doing this is to reduce the number of times a frequently-modified piece of data must be written back to RAM.

Write-through cache is a bit simpler to implement; for this reason it is most common. Write-back cache is a bit trickier to implement; in addition to storing the actual data, it is necessary to maintain some sort of mechanism capable of flagging the cached data as clean (the data in cache is the same as the data in RAM), or dirty (the data in cache has been modified, meaning that the data in RAM is no longer current). It is also necessary to implement a way of periodically flushing dirty cache entries back to RAM.

4.2.2.1. Cache Levels

Cache subsystems in present-day computer designs may be multi-level; that is, there might be more than one set of cache between the CPU and main memory. The cache levels are often numbered, with lower numbers being closer to the CPU. Many systems have two cache levels:

- L1 cache is often located directly on the CPU chip itself and runs at the same speed as the CPU
- L2 cache is often part of the CPU module, runs at CPU speeds (or nearly so), and is usually a bit larger and slower than L1 cache

Some systems (normally high-performance servers) also have L3 cache, which is usually part of the system motherboard. As might be expected, L3 cache would be larger (and most likely slower) than L2 cache.

In either case, the goal of all cache subsystems -- whether single- or multi-level -- is to reduce the average access time to the RAM.

4.2.3. Main Memory -- RAM

RAM makes up the bulk of electronic storage on present-day computers. It is used as storage for both data and programs while those data and programs are in use. The speed of RAM in most systems today lies between the speed of cache memory and that of hard drives, and is much closer to the former than the latter.

The basic operation of RAM is actually quite straightforward. At the lowest level, there are the RAM chips -- integrated circuits that do the actual "remembering." These chips have four types of connections to the outside world:

- Power connections (to operate the circuitry within the chip)
- Data connections (to enable the transfer of data into or out of the chip)
- Read/Write connections (to control whether data is to be stored into or retrieved from the chip)
- Address connections (to determine where in the chip the data should be read/written)

Here are the steps required to store data in RAM:

1. The data to be stored is presented to the data connections.

2. The address at which the data is to be stored is presented to the address connections.

3. The read/write connection is set to write mode.

Retrieving data is just as straightforward:

1. The address of the desired data is presented to the address connections.

2. The read/write connection is set to read mode.

3. The desired data is read from the data connections.

While these steps seem simple, they take place at very high speeds, with the time spent on each step measured in nanoseconds.

Nearly all RAM chips created today are sold as *modules*. Each module consists of a number of individual RAM chips attached to a small circuit board. The mechanical and electrical layout of the module adheres to various industry standards, making it possible to purchase memory from a variety of vendors.

> **Note**
>
> The main benefit to a system using industry-standard RAM modules is that it tends to keep the cost of RAM low, due to the ability to purchase the modules from more than just the system manufacturer.
>
> Although most computers use industry-standard RAM modules, there are exceptions. Most notable are laptops (and even here some standardization is starting to take hold) and high-end servers. However, even in these instances, it is likely that third-party RAM modules are available, assuming the system is relatively popular and is not a completely new design.

4.2.4. Hard Drives

All the technologies discussed so far are *volatile* in nature. In other words, data contained in volatile storage is lost when the power is turned off.

Hard drives, on the other hand, are *non-volatile* -- the data they contain remains there, even after the power is removed. Because of this, hard drives occupy a special place in the storage spectrum. Their non-volatile nature makes them ideal for storing programs and data for longer-term use. Another unique aspect to hard drives is that, unlike RAM and cache memory, it is not possible to execute programs directly when they are stored on hard drives; instead, they must first be read into RAM.

Also different from cache and RAM is the speed of data storage and retrieval; hard drives are at least an order of magnitude slower than the all-electronic technologies used for cache and RAM. The difference in speed is due mainly to their electromechanical nature. There are four distinct phases taking place during each data transfer to or from a hard drive. The following list illustrates these phases, along with the time it would take a typical high-performance drive, on average, to complete each:

- Access arm movement (5.5 milliseconds)

- Disk rotation (.1 milliseconds)

- Heads reading/writing data (.00014 milliseconds)

- Data transfer to/from the drive's electronics (.003 Milliseconds)

Of these, only the last phase is not dependent on any mechanical operation.

Note

Although there is much more to learn about hard drives, disk storage technologies are discussed in more depth in *Chapter 5, Managing Storage*. For the time being, it is only necessary to keep in mind the huge speed difference between RAM and disk-based technologies and that their storage capacity usually exceeds that of RAM by a factor of at least 10, and often by 100 or more.

4.2.5. Off-Line Backup Storage

Off-line backup storage takes a step beyond hard drive storage in terms of capacity (higher) and speed (slower). Here, capacities are effectively limited only by your ability to procure and store the removable media.

The actual technologies used in these devices varies widely. Here are the more popular types:

- Magnetic tape
- Optical disk

Of course, having removable media means that access times become even longer, particularly when the desired data is on media not currently loaded in the storage device. This situation is alleviated somewhat by the use of robotic devices capable of automatically loading and unloading media, but the media storage capacities of such devices are still finite. Even in the best of cases, access times are measured in seconds, which is much longer than the relatively slow multi-millisecond access times typical for a high-performance hard drive.

Now that we have briefly studied the various storage technologies in use today, let us explore basic virtual memory concepts.

4.3. Basic Virtual Memory Concepts

While the technology behind the construction of the various modern-day storage technologies is truly impressive, the average system administrator does not need to be aware of the details. In fact, there is really only one fact that system administrators should always keep in mind:

There is never enough RAM.

While this truism might at first seem humorous, many operating system designers have spent a great deal of time trying to reduce the impact of this very real shortage. They have done so by implementing *virtual memory* -- a way of combining RAM with slower storage to give a system the appearance of having more RAM than is actually installed.

4.3.1. Virtual Memory in Simple Terms

Let us start with a hypothetical application. The machine code making up this application is 10000 bytes in size. It also requires another 5000 bytes for data storage and I/O buffers. This means that, to run this application, there must be 15000 bytes of RAM available; even one byte less, and the application would not be able to run.

This 15000 byte requirement is known as the application's *address space*. It is the number of unique addresses needed to hold both the application and its data. In the first computers, the amount of available RAM had to be greater than the address space of the largest application to be run; otherwise, the application would fail with an "out of memory" error.

A later approach known as *overlaying* attempted to alleviate the problem by allowing programmers to dictate which parts of their application needed to be memory-resident at any given time. In this way, code only required once for initialization purposes could be written over (overlayed) with code that would be used later. While overlays did ease memory shortages, it was a very complex and error-prone process. Overlays also failed to address the issue of system-wide memory shortages at runtime. In other words, an overlayed program may require less memory to run than a program that is not overlayed, but if the system still does not have sufficient memory for the overlayed program, the end result is the same -- an out of memory error.

With virtual memory, the concept of an application's address space takes on a different meaning. Rather than concentrating on how *much* memory an application needs to run, a virtual memory operating system continually attempts to find the answer to the question, "how *little* memory does an application need to run?"

While it at first appears that our hypothetical application requires the full 15000 bytes to run, think back to our discussion in *Section 4.1, “Storage Access Patterns”* -- memory access tends to be sequential and localized. Because of this, the amount of memory required to execute the application at any given time is less than 15000 bytes -- usually a lot less. Consider the types of memory accesses required to execute a single machine instruction:

- The instruction is read from memory.
- The data required by the instruction is read from memory.
- After the instruction completes, the results of the instruction are written back to memory.

The actual number of bytes necessary for each memory access varies according to the CPU's architecture, the actual instruction, and the data type. However, even if one instruction required 100 bytes of memory for each type of memory access, the 300 bytes required is still much less than the application's entire 15000-byte address space. If a way could be found to keep track of an application's memory requirements as the application runs, it would be possible to keep the application running while using less memory than its address space would otherwise dictate.

But that leaves one question:

If only part of the application is in memory at any given time, where is the rest of it?

4.3.2. Backing Store -- the Central Tenet of Virtual Memory

The short answer to this question is that the rest of the application remains on disk. In other words, disk acts as the *backing store* for RAM; a slower, larger storage medium acting as a "backup" for a much faster, smaller storage medium. This might at first seem to be a very large performance problem in the making -- after all, disk drives are so much slower than RAM.

While this is true, it is possible to take advantage of the sequential and localized access behavior of applications and eliminate most of the performance implications of using disk drives as backing store for RAM. This is done by structuring the virtual memory subsystem so that it attempts to ensure that those parts of the application currently needed -- or likely to be needed in the near future -- are kept in RAM only for as long as they are actually needed.

In many respects this is similar to the relationship between cache and RAM: making a small amount of fast storage combined with a large amount of slow storage act just like a large amount of fast storage.

With this in mind, let us explore the process in more detail.

4.4. Virtual Memory: The Details

First, we must introduce a new concept: *virtual address space*. Virtual address space is the maximum amount of address space available to an application. The virtual address space varies according to the system's architecture and operating system. Virtual address space depends on the architecture because it is the architecture that defines how many bits are available for addressing purposes. Virtual address space also depends on the operating system because the manner in which the operating system was implemented may introduce additional limits over and above those imposed by the architecture.

The word "virtual" in virtual address space means this is the total number of uniquely-addressable memory locations available to an application, but *not* the amount of physical memory either installed in the system, or dedicated to the application at any given time.

In the case of our example application, its virtual address space is 15000 bytes.

To implement virtual memory, it is necessary for the computer system to have special memory management hardware. This hardware is often known as an *MMU* (Memory Management Unit). Without an MMU, when the CPU accesses RAM, the actual RAM locations never change -- memory address 123 is always the same physical location within RAM.

However, with an MMU, memory addresses go through a translation step prior to each memory access. This means that memory address 123 might be directed to physical address 82043 at one time, and physical address 20468 another time. As it turns out, the overhead of individually tracking the virtual to physical translations for billions of bytes of memory would be too great. Instead, the MMU divides RAM into *pages* -- contiguous sections of memory of a set size that are handled by the MMU as single entities.

Keeping track of these pages and their address translations might sound like an unnecessary and confusing additional step. However, it is crucial to implementing virtual memory. For that reason, consider the following point.

Taking our hypothetical application with the 15000 byte virtual address space, assume that the application's first instruction accesses data stored at address 12374. However, also assume that our computer only has 12288 bytes of physical RAM. What happens when the CPU attempts to access address 12374?

What happens is known as a *page fault*.

4.4.1. Page Faults

A page fault is the sequence of events occurring when a program attempts to access data (or code) that is in its address space, but is not currently located in the system's RAM. The operating system must handle page faults by somehow making the accessed data memory resident, allowing the program to continue operation as if the page fault had never occurred.

In the case of our hypothetical application, the CPU first presents the desired address (12374) to the MMU. However, the MMU has no translation for this address. So, it interrupts the CPU and causes software, known as a page fault handler, to be executed. The page fault handler then determines what must be done to resolve this page fault. It can:

- Find where the desired page resides on disk and read it in (this is normally the case if the page fault is for a page of code)
- Determine that the desired page is already in RAM (but not allocated to the current process) and reconfigure the MMU to point to it

- Point to a special page containing only zeros, and allocate a new page for the process only if the process ever attempts to write to the special page (this is called a *copy on write* page, and is often used for pages containing zero-initialized data)

- Get the desired page from somewhere else (which is discussed in more detail later)

While the first three actions are relatively straightforward, the last one is not. For that, we need to cover some additional topics.

4.4.2. The Working Set

The group of physical memory pages currently dedicated to a specific process is known as the *working set* for that process. The number of pages in the working set can grow and shrink, depending on the overall availability of pages on a system-wide basis.

The working set grows as a process page faults. The working set shrinks as fewer and fewer free pages exist. To keep from running out of memory completely, pages must be removed from process's working sets and turned into free pages, available for later use. The operating system shrinks processes' working sets by:

- Writing modified pages to a dedicated area on a mass storage device (usually known as *swapping* or *paging* space)

- Marking unmodified pages as being free (there is no need to write these pages out to disk as they have not changed)

To determine appropriate working sets for all processes, the operating system must track usage information for all pages. In this way, the operating system determines which pages are actively being used (and must remain memory resident) and which pages are not (and therefore, can be removed from memory.) In most cases, some sort of least-recently used algorithm determines which pages are eligible for removal from process working sets.

4.4.3. Swapping

While swapping (writing modified pages out to the system swap space) is a normal part of a system's operation, it is possible to experience too much swapping. The reason to be wary of excessive swapping is that the following situation can easily occur, over and over again:

- Pages from a process are swapped

- The process becomes runnable and attempts to access a swapped page

- The page is faulted back into memory (most likely forcing some other processes' pages to be swapped out)

- A short time later, the page is swapped out again

If this sequence of events is widespread, it is known as *thrashing* and is indicative of insufficient RAM for the present workload. Thrashing is extremely detrimental to system performance, as the CPU and I/O loads that can be generated in such a situation quickly outweigh the load imposed by a system's real work. In extreme cases, the system may actually do no useful work, spending all its resources moving pages to and from memory.

4.5. Virtual Memory Performance Implications

While virtual memory makes it possible for computers to more easily handle larger and more complex applications, as with any powerful tool, it comes at a price. The price in this case is one of performance -- a virtual memory operating system has a lot more to do than an operating system incapable of supporting virtual memory. This means that performance is never as good with virtual memory as it is when the same application is 100% memory-resident.

However, this is no reason to throw up one's hands and give up. The benefits of virtual memory are too great to do that. And, with a bit of effort, good performance is possible. The thing that must be done is to examine those system resources impacted by heavy use of the virtual memory subsystem.

4.5.1. Worst Case Performance Scenario

For a moment, take what you have read in this chapter and consider what system resources are used by extremely heavy page fault and swapping activity:

- RAM -- It stands to reason that available RAM is low (otherwise there would be no need to page fault or swap).
- Disk -- While disk space might not be impacted, I/O bandwidth (due to heavy paging and swapping) would be.
- CPU -- The CPU is expending cycles doing the processing required to support memory management and setting up the necessary I/O operations for paging and swapping.

The interrelated nature of these loads makes it easy to understand how resource shortages can lead to severe performance problems.

All it takes is a system with too little RAM, heavy page fault activity, and a system running near its limit in terms of CPU or disk I/O. At this point, the system is thrashing, with poor performance the inevitable result.

4.5.2. Best Case Performance Scenario

At best, the overhead from virtual memory support presents a minimal additional load to a well-configured system:

- RAM -- Sufficient RAM for all working sets with enough left over to handle any page faults[2]
- Disk -- Because of the limited page fault activity, disk I/O bandwidth would be minimally impacted
- CPU -- The majority of CPU cycles are dedicated to actually running applications, instead of running the operating system's memory management code

From this, the overall point to keep in mind is that the performance impact of virtual memory is minimal when it is used as little as possible. This means the primary determinant of good virtual memory subsystem performance is having enough RAM.

Next in line (but much lower in relative importance) are sufficient disk I/O and CPU capacity. However, keep in mind that these resources only help the system performance degrade more gracefully from heavy faulting and swapping; they do little to help the virtual memory subsystem performance (although they obviously can play a major role in overall system performance).

4.6. Red Hat Enterprise Linux-Specific Information

Due to the inherent complexity of being a demand-paged virtual memory operating system, monitoring memory-related resources under Red Hat Enterprise Linux can be confusing. Therefore, it is best to start with the more straightforward tools, and work from there.

Using **`free`**, it is possible to get a concise (if somewhat simplistic) overview of memory and swap utilization. Here is an example:

```
 total used free shared buffers cached Mem: 1288720 361448 927272 0 27844 187632 -/+ buffers/
cache: 145972 1142748 Swap: 522104 0 522104
```

We note that this system has 1.2GB of RAM, of which only about 350MB is actually in use. As expected for a system with this much free RAM, none of the 500MB swap partition is in use.

Contrast that example with this one:

```
 total used free shared buffers cached Mem: 255088 246604 8484 0 6492 111320 -/+ buffers/
cache: 128792 126296 Swap: 530136 111308 418828
```

This system has about 256MB of RAM, the majority of which is in use, leaving only about 8MB free. Over 100MB of the 512MB swap partition is in use. Although this system is certainly more limited in terms of memory than the first system, to determine if this memory limitation is causing performance problems we must dig a bit deeper.

Although more cryptic than **`free`**, **`vmstat`** has the benefit of displaying more than memory utilization statistics. Here is the output from **`vmstat 1 10`**:

```
 procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu---- r b swpd free
 buff cache si so bi bo in cs us sy id wa 2 0 111304 9728 7036 107204 0 0 6 10 120 24 10 2 89
 0 2 0 111304 9728 7036 107204 0 0 0 0 526 1653 96 4 0 0 1 0 111304 9616 7036 107204 0 0 0 0
 552 2219 94 5 1 0 1 0 111304 9616 7036 107204 0 0 0 0 624 699 98 2 0 0 2 0 111304 9616 7052
 107204 0 0 0 48 603 1466 95 5 0 0 3 0 111304 9620 7052 107204 0 0 0 0 768 932 90 4 6 0 3 0
 111304 9440 7076 107360 92 0 244 0 820 1230 85 9 6 0 2 0 111304 9276 7076 107368 0 0 0 0 832
 1060 87 6 7 0 3 0 111304 9624 7092 107372 0 0 16 0 813 1655 93 5 2 0 2 0 111304 9624 7108
 107372 0 0 0 972 1189 1165 68 9 23 0
```

During this 10-second sample, the amount of free memory (the **`free`** field) varies somewhat, and there is a bit of swap-related I/O (the **`si`** and **`so`** fields), but overall this system is running well. It is doubtful, however, how much additional workload it could handle, given the current memory utilization.

When researching memory-related issues, it is often necessary to determine how the Red Hat Enterprise Linux virtual memory subsystem is making use of system memory. By using **`sar`**, it is possible to examine this aspect of system performance in much more detail.

By reviewing the **`sar -r`** report, we can examine memory and swap utilization more closely:

```
 Linux 2.4.20-1.1931.2.231.2.10.ent (pigdog.example.com) 07/22/2003 12:00:01 AM kbmemfree
 kbmemused %memused kbmemshrd kbbuffers kbcached 12:10:00 AM 240468 1048252 81.34 0 133724
 485772 12:20:00 AM 240508 1048212 81.34 0 134172 485600 … 08:40:00 PM 934132 354588 27.51 0
 26080 185364 Average: 324346 964374 74.83 0 96072 467559
```

The **`kbmemfree`** and **`kbmemused`** fields show the typical free and used memory statistics, with the percentage of memory used displayed in the **`%memused`** field. The **`kbbuffers`** and **`kbcached`** fields show how many kilobytes of memory are allocated to buffers and the system-wide data cache.

The **`kbmemshrd`** field is always zero for systems (such as Red Hat Enterprise Linux) using the 2.4 Linux kernel.

The lines for this report have been truncated to fit on the page. Here is the remainder of each line, with the timestamp added to the left to make reading easier:

```
12:00:01 AM kbswpfree kbswpused %swpused 12:10:00 AM 522104 0 0.00 12:20:00 AM 522104 0 0.00
… 08:40:00 PM 522104 0 0.00 Average: 522104 0 0.00
```

For swap utilization, the **`kbswpfree`** and **`kbswpused`** fields show the amount of free and used swap space, in kilobytes, with the **`%swpused`** field showing the swap space used as a percentage.

To learn more about the swapping activity taking place, use the **`sar -W`** report. Here is an example:

```
Linux 2.4.20-1.1931.2.231.2.10.entsmp (raptor.example.com) 07/22/2003 12:00:01 AM pswpin/s
pswpout/s 12:10:01 AM 0.15 2.56 12:20:00 AM 0.00 0.00 … 03:30:01 PM 0.42 2.56 Average: 0.11
0.37
```

Here we notice that, on average, there were three times fewer pages being brought in from swap (**`pswpin/s`**) as there were going out to swap (**`pswpout/s`**).

To better understand how pages are being used, refer to the **`sar -B`** report:

```
Linux 2.4.20-1.1931.2.231.2.10.entsmp (raptor.example.com) 07/22/2003 12:00:01 AM pgpgin/s
pgpgout/s activepg inadtypg inaclnpg inatarpg 12:10:00 AM 0.03 8.61 195393 20654 30352 49279
12:20:00 AM 0.01 7.51 195385 20655 30336 49275 … 08:40:00 PM 0.00 7.79 71236 1371 6760 15873
Average: 201.54 201.54 169367 18999 35146 44702
```

Here we can determine how many blocks per second are paged in from disk (**`pgpgin/s`**) and paged out to disk (**`pgpgout/s`**). These statistics serve as a barometer of overall virtual memory activity.

However, more knowledge can be gained by examining the other fields in this report. The Red Hat Enterprise Linux kernel marks all pages as either active or inactive. As the names imply, active pages are currently in use in some manner (as process or buffer pages, for example), while inactive pages are not. This example report shows that the list of active pages (the **`activepg`** field) averages approximately 660MB[3].

The remainder of the fields in this report concentrate on the inactive list -- pages that, for one reason or another, have not recently been used. The **`inadtypg`** field shows how many inactive pages are *dirty* (modified) and may need to be written to disk. The **`inaclnpg`** field, on the other hand, shows how many inactive pages are *clean* (unmodified) and do not need to be written to disk.

The **`inatarpg`** field represents the desired size of the inactive list. This value is calculated by the Linux kernel and is sized such that the inactive list remains large enough to act as a pool for page replacement purposes.

For additional insight into page status (specifically, how often pages change status), use the **`sar -R`** report. Here is a sample report:

```
Linux 2.4.20-1.1931.2.231.2.10.entsmp (raptor.example.com) 07/22/2003 12:00:01 AM frmpg/s
shmpg/s bufpg/s campg/s 12:10:00 AM -0.10 0.00 0.12 -0.07 12:20:00 AM 0.02 0.00 0.19 -0.07 …
08:50:01 PM -3.19 0.00 0.46 0.81 Average: 0.01 0.00 -0.00 -0.00
```

[3] The page size under Red Hat Enterprise Linux on the x86 system used in this example is 4096 bytes. Systems based on other architectures may have different page sizes.

The statistics in this particular **`sar`** report are unique, in that they may be positive, negative, or zero. When positive, the value indicates the rate at which pages of this type are increasing. When negative, the value indicates the rate at which pages of this type are decreasing. A value of zero indicates that pages of this type are neither increasing or decreasing.

In this example, the last sample shows slightly over three pages per second being allocated from the list of free pages (the **`frmpg/s`** field) and nearly 1 page per second added to the page cache (the **`campg/s`** field). The list of pages used as buffers (the **`bufpg/s`** field) gained approximately one page every two seconds, while the shared memory page list (the **`shmpg/s`** field) neither gained nor lost any pages.

4.7. Additional Resources

This section includes various resources that can be used to learn more about resource monitoring and the Red Hat Enterprise Linux-specific subject matter discussed in this chapter.

4.7.1. Installed Documentation

The following resources are installed in the course of a typical Red Hat Enterprise Linux installation and can help you learn more about the subject matter discussed in this chapter.

- **`free(1)`** man page -- Learn how to display free and used memory statistics.
- **`vmstat(8)`** man page -- Learn how to display a concise overview of process, memory, swap, I/O, system, and CPU utilization.
- **`sar(1)`** man page -- Learn how to produce system resource utilization reports.
- **`sa2(8)`** man page -- Learn how to produce daily system resource utilization report files.

4.7.2. Useful Websites

- *http://people.redhat.com/alikins/system_tuning.html* -- System Tuning Info for Linux Servers. A stream-of-consciousness approach to performance tuning and resource monitoring for servers.
- *http://www.linuxjournal.com/article.php?sid=2396* -- Performance Monitoring Tools for Linux. This Linux Journal page is geared more toward the administrator interested in writing a customized performance graphing solution. Written several years ago, some of the details may no longer apply, but the overall concept and execution are sound.

4.7.3. Related Books

The following books discuss various issues related to resource monitoring, and are good resources for Red Hat Enterprise Linux system administrators:

- The *System Administrators Guide*; Red Hat, Inc -- Includes a chapter on many of the resource monitoring tools described here.
- *Linux Performance Tuning and Capacity Planning* by Jason R. Fink and Matthew D. Sherer; Sams -- Provides more in-depth overviews of the resource monitoring tools presented here and includes others that might be appropriate for more specific resource monitoring needs.
- *Red Hat Linux Security and Optimization* by Mohammed J. Kabir; Red Hat Press -- Approximately the first 150 pages of this book discuss performance-related issues. This includes chapters dedicated to performance issues specific to network, Web, email, and file servers.

- *Linux Administration Handbook* by Evi Nemeth, Garth Snyder, and Trent R. Hein; Prentice Hall -- Provides a short chapter similar in scope to this book, but includes an interesting section on diagnosing a system that has suddenly slowed down.
- *Linux System Administration: A User's Guide* by Marcel Gagne; Addison Wesley Professional -- Contains a small chapter on performance monitoring and tuning.
- *Essential System Administration* (3rd Edition) by Aeleen Frisch; O'Reilly &Associates -- The chapter on managing system resources contains good overall information, with some Linux specifics included.
- *System Performance Tuning* (2nd Edition) by Gian-Paolo D. Musumeci and Mike Loukides; O'Reilly &Associates -- Although heavily oriented toward more traditional UNIX implementations, there are many Linux-specific references throughout the book.

Managing Storage

If there is one thing that takes up the majority of a system administrator's day, it would have to be storage management. It seems that disks are always running out of free space, becoming overloaded with too much I/O activity, or failing unexpectedly. Therefore, it is vital to have a solid working knowledge of disk storage in order to be a successful system administrator.

5.1. An Overview of Storage Hardware

Before managing storage, it is first necessary to understand the hardware on which data is stored. Unless you have at least some knowledge about mass storage device operation, you may find yourself in a situation where you have a storage-related problem, but you lack the background knowledge necessary to interpret what you are seeing. By gaining some insight into how the underlying hardware operates, you should be able to more easily determine whether your computer's storage subsystem is operating properly.

The vast majority of all mass-storage devices use some sort of rotating media and supports the random access of data on that media. This means that the following components are present in some form within nearly every mass storage device:

- Disk platters
- Data reading/writing device
- Access arms

The following sections explore each of these components in more detail.

5.1.1. Disk Platters

The rotating media used by nearly all mass storage devices are in the form of one or more flat, circularly-shaped platters. The platter may be composed of any number of different materials, such aluminum, glass, and polycarbonate.

The surface of each platter is treated in such a way as to enable data storage. The exact nature of the treatment depends on the data storage technology to be used. The most common data storage technology is based on the property of magnetism; in these cases the platters are covered with a compound that exhibits good magnetic characteristics.

Another common data storage technology is based on optical principles; in these cases, the platters are covered with materials whose optical properties can be modified, thereby allowing data to be stored optically[1].

No matter what data storage technology is in use, the disk platters are spun, causing their entire surface to sweep past another component -- the data reading/writing device.

5.1.2. Data reading/writing device

The data reading/writing device is the component that takes the bits and bytes on which a computer system operates and turns them into the magnetic or optical variations necessary to interact with the materials coating the surface of the disk platters.

[1] Some optical devices -- notably CD-ROM drives -- use somewhat different approaches to data storage; these differences are pointed out at the appropriate points within the chapter.

Sometimes the conditions under which these devices must operate are challenging. For instance, in magnetically-based mass storage the read/write devices (known as *heads*) must be very close to the surface of the platter. However, if the head and the surface of the disk platter were to touch, the resulting friction would do severe damage to both the head and the platter. Therefore, the surfaces of both the head and the platter are carefully polished, and the head uses air pressure developed by the spinning platters to float over the platter's surface, "flying" at an altitude less than the thickness of a human hair. This is why magnetic disk drives are sensitive to shock, sudden temperature changes, and any airborne contamination.

The challenges faced by optical heads are somewhat different than for magnetic heads -- here, the head assembly must remain at a relatively constant distance from the surface of the platter. Otherwise, the lenses used to focus on the platter does not produce a sufficiently sharp image.

In either case, the heads use a very small amount of the platter's surface area for data storage. As the platter spins below the heads, this surface area takes the form of a very thin circular line.

If this was how mass storage devices worked, it would mean that over 99% of the platter's surface area would be wasted. Additional heads could be mounted over the platter, but to fully utilize the platter's surface area more than a thousand heads would be necessary. What is required is some method of moving the head over the surface of the platter.

5.1.3. Access Arms

By using a head attached to an arm that is capable of sweeping over the platter's entire surface, it is possible to fully utilize the platter for data storage. However, the access arm must be capable of two things:

- Moving very quickly
- Moving very precisely

The access arm must move as quickly as possible, because the time spent moving the head from one position to another is wasted time. That is because no data can be read or written until the access arm stops moving[2].

The access arm must be able to move with great precision because, as stated earlier, the surface area used by the heads is very small. Therefore, to efficiently use the platter's storage capacity, it is necessary to move the heads only enough to ensure that any data written in the new position does not overwrite data written at a previous position. This has the affect of conceptually dividing the platter's surface into a thousand or more concentric "rings" or *tracks*. Movement of the access arm from one track to another is often referred to as *seeking*, and the time it takes the access arms to move from one track to another is known as the *seek time*.

Where there are multiple platters (or one platter with both surfaces used for data storage), the arms for each surface are stacked, allowing the same track on each surface to be accessed simultaneously. If the tracks for each surface could be visualized with the access stationary over a given track, they would appear to be stacked one on top of another, making up a cylindrical shape; therefore, the set of tracks accessible at a certain position of the access arms are known as a *cylinder*.

[2] In some optical devices (such as CD-ROM drives), the access arm is continually moving, causing the head assembly to describe a spiral path over the surface of the platter. This is a fundamental difference in how the storage medium is used and reflects the CD-ROM's origins as a medium for music storage, where continuous data retrieval is a more common operation than searching for a specific data point.

5.2. Storage Addressing Concepts

The configuration of disk platters, heads, and access arms makes it possible to position the head over any part of any surface of any platter in the mass storage device. However, this is not sufficient; to use this storage capacity, we must have some method of giving addresses to uniform-sized parts of the available storage.

There is one final aspect to this process that is required. Consider all the tracks in the many cylinders present in a typical mass storage device. Because the tracks have varying diameters, their circumference also varies. Therefore, if storage was addressed only to the track level, each track would have different amounts of data -- track #0 (being near the center of the platter) might hold 10,827 bytes, while track #1,258 (near the outside edge of the platter) might hold 15,382 bytes.

The solution is to divide each track into multiple *sectors* or *blocks* of consistently-sized (often 512 bytes) segments of storage. The result is that each track contains a set number[3] of sectors.

A side effect of this is that every track contains unused space -- the space between the sectors. Despite the constant number of sectors in each track, the amount of unused space varies -- relatively little unused space in the inner tracks, and a great deal more unused space in the outer tracks. In either case, this unused space is wasted, as data cannot be stored on it.

However, the advantage offsetting this wasted space is that effectively addressing the storage on a mass storage device is now possible. In fact, there are two methods of addressing -- geometry-based addressing and block-based addressing.

5.2.1. Geometry-Based Addressing

The term *geometry-based addressing* refers to the fact that mass storage devices actually store data at a specific physical spot on the storage medium. In the case of the devices being described here, this refers to three specific items that define a specific point on the device's disk platters:

- Cylinder
- Head
- Sector

The following sections describe how a hypothetical address can describe a specific physical location on the storage medium.

5.2.1.1. Cylinder

As stated earlier, the cylinder denotes a specific position of the access arm (and therefore, the read/write heads). By specifying a particular cylinder, we are eliminating all other cylinders, reducing our search to only one track for each surface in the mass storage device.

Table 5.1. Storage Addressing

Cylinder	Head	Sector
1014	*X*	*X*

In *Table 5.1, “Storage Addressing”*, the first part of a geometry-based address has been filled in. Two more components to this address -- the head and sector -- remain undefined.

[3] While early mass storage devices used the same number of sectors for every track, later devices divided the range of cylinders into different "zones," with each zone having a different number of sectors per track. The reason for this is to take advantage of the additional space between sectors in the outer cylinders, where there is more unused space between sectors.

5.2.1.2. Head

Although in the strictest sense we are selecting a particular disk platter, because each surface has a read/write head dedicated to it, it is easier to think in terms of interacting with a specific head. In fact, the device's underlying electronics actually select one head and -- deselecting the rest -- only interact with the selected head for the duration of the I/O operation. All other tracks that make up the current cylinder have now been eliminated.

Table 5.2. Storage Addressing

Cylinder	Head	Sector
1014	2	*X*

In *Table 5.2, “Storage Addressing”*, the first two parts of a geometry-based address have been filled in. One final component to this address -- the sector -- remains undefined.

5.2.1.3. Sector

By specifying a particular sector, we have completed the addressing, and have uniquely identified the desired block of data.

Table 5.3. Storage Addressing

Cylinder	Head	Sector
1014	2	12

In *Table 5.3, “Storage Addressing”*, the complete geometry-based address has been filled in. This address identifies the location of one specific block out of all the other blocks on this device.

5.2.1.4. Problems with Geometry-Based Addressing

While geometry-based addressing is straightforward, there is an area of ambiguity that can cause problems. The ambiguity is in numbering the cylinders, heads, and sectors.

It is true that each geometry-based address uniquely identifies one specific data block, but that only applies if the numbering scheme for the cylinders, heads, and sectors is not changed. If the numbering scheme changes (such as when the hardware/software interacting with the storage device changes), then the mapping between geometry-based addresses and their corresponding data blocks can change, making it impossible to access the desired data.

Because of this potential for ambiguity, a different approach to addressing was developed. The next section describes it in more detail.

5.2.2. Block-Based Addressing

Block-based addressing is much more straightforward than geometry-based addressing. With block-based addressing, every data block is given a unique number. This number is passed from the computer to the mass storage device, which then internally performs the conversion to the geometry-based address required by the device's control circuitry.

Because the conversion to a geometry-based address is always done by the device itself, it is always consistent, eliminating the problem inherent with giving the device geometry-based addressing.

5.3. Mass Storage Device Interfaces

Every device used in a computer system must have some means of attaching to that computer system. This attachment point is known as an *interface*. Mass storage devices are no different -- they have interfaces too. It is important to know about interfaces for two main reasons:

- There are many different (mostly incompatible) interfaces
- Different interfaces have different performance and price characteristics

Unfortunately, there is no single universal device interface and not even a single mass storage device interface. Therefore, system administrators must be aware of the interface(s) supported by their organization's systems. Otherwise, there is a real risk of purchasing the wrong hardware when a system upgrade is planned.

Different interfaces have different performance capabilities, making some interfaces more suitable for certain environments than others. For example, interfaces capable of supporting high-speed devices are more suitable for server environments, while slower interfaces would be sufficient for light desktop usage. Such differences in performance also lead to differences in price, meaning that -- as always -- you get what you pay for. High-performance computing does not come cheaply.

5.3.1. Historical Background

Over the years there have been many different interfaces created for mass storage devices. Some have fallen by the wayside, and some are still in use today. However, the following list is provided to give an idea of the scope of interface development over the past thirty years and to provide perspective on the interfaces in use today.

FD-400

An interface originally designed for the original 8-inch floppy disk drives in the mid-70s. Used a 44-conductor cable with an circuit board edge connector that supplied both power and data.

SA-400

Another floppy disk drive interface (this time originally developed in the late-70s for the then-new 5.25 inch floppy drive). Used a 34-conductor cable with a standard socket connector. A slightly modified version of this interface is still used today for 5.25 inch floppy and 3.5 inch diskette drives.

IPI

Standing for Intelligent Peripheral Interface, this interface was used on the 8 and 14-inch disk drives deployed on minicomputers of the 1970s.

SMD

A successor to IPI, SMD (stands for Storage Module Device) was used on 8 and 14-inch minicomputer hard drives in the 70s and 80s.

ST506/412

A hard drive interface dating from the early 80s. Used in many personal computers of the day, this interface used two cables -- one 34-conductor and one 20-conductor.

ESDI

Standing for Enhanced Small Device Interface, this interface was considered a successor to ST506/412 with faster transfer rates and larger supported drive sizes. Dating from the mid-80s, ESDI used the same two-cable connection scheme of its predecessor.

There were also proprietary interfaces from the larger computer vendors of the day (IBM and DEC, primarily). The intent behind the creation of these interfaces was to attempt to protect the extremely lucrative peripherals business for their computers. However, due to their proprietary nature, the devices compatible with these interfaces were more expensive than equivalent non-proprietary devices. Because of this, these interfaces failed to achieve any long-term popularity.

While proprietary interfaces have largely disappeared, and the interfaces described at the start of this section no longer have much (if any) market share, it is important to know about these no-longer-used interfaces, as they prove one point -- nothing in the computer industry remains constant for long. Therefore, always be on the lookout for new interface technologies; one day you might find that one of them may prove to be a better match for your needs than the more traditional offerings you currently use.

5.3.2. Present-Day Industry-Standard Interfaces

Unlike the proprietary interfaces mentioned in the previous section, some interfaces were more widely adopted, and turned into industry standards. Two interfaces in particular have made this transition and are at the heart of today's storage industry:

- IDE/ATA
- SCSI

5.3.2.1. IDE/ATA

IDE stands for Integrated Drive Electronics. This interface originated in the late 80s, and uses a 40-pin connector.

Note

Actually, the proper name for this interface is the "AT Attachment" interface (or ATA), but use of the term "IDE" (which actually refers to an ATA-compatible mass storage device) is still used to some extent. However, the remainder of this section uses the interface's proper name -- ATA.

ATA implements a bus topology, with each bus supporting two mass storage devices. These two devices are known as the *master* and the *slave*. These terms are misleading, as it implies some sort of relationship between the devices; that is not the case. The selection of which device is the master and which is the slave is normally selected through the use of jumper blocks on each device.

Note

A more recent innovation is the introduction of *cable select* capabilities to ATA. This innovation requires the use of a special cable, an ATA controller, and mass storage devices that support cable select (normally through a "cable select" jumper setting). When properly configured, cable select eliminates the need to change jumpers when moving devices; instead, the device's position on the ATA cable denotes whether it is master or slave.

A variation of this interface illustrates the unique ways in which technologies can be mixed and also introduces our next industry-standard interface. *ATAPI* is a variation of the ATA interface and stands for AT Attachment Packet Interface. Used primarily by CD-ROM drives, ATAPI adheres to the electrical and mechanical aspects of the ATA interface but uses the communication protocol from the next interface discussed -- SCSI.

5.3.2.2. SCSI

Formally known as the Small Computer System Interface, SCSI as it is known today originated in the early 80s and was declared a standard in 1986. Like ATA, SCSI makes use of a bus topology. However, there the similarities end.

Using a bus topology means that every device on the bus must be uniquely identified somehow. While ATA supports only two different devices for each bus and gives each one a specific name, SCSI does this by assigning each device on a SCSI bus a unique numeric address or *SCSI ID*. Each device on a SCSI bus must be configured (usually by jumpers or switches[4]) to respond to its SCSI ID.

Before continuing any further in this discussion, it is important to note that the SCSI standard does not represent a single interface, but a family of interfaces. There are several areas in which SCSI varies:

- Bus width
- Bus speed
- Electrical characteristics

The original SCSI standard described a bus topology in which eight lines in the bus were used for data transfer. This meant that the first SCSI devices could transfer data one byte at a time. In later years, the standard was expanded to permit implementations where sixteen lines could be used, doubling the amount of data that devices could transfer. The original "8-bit" SCSI implementations were then referred to as *narrow* SCSI, while the newer 16-bit implementations were known as *wide* SCSI.

Originally, the bus speed for SCSI was set to 5MHz, permitting a 5MB/second transfer rate on the original 8-bit SCSI bus. However, subsequent revisions to the standard doubled that speed to 10MHz, resulting in 10MB/second for narrow SCSI and 20MB/second for wide SCSI. As with the bus width, the changes in bus speed received new names, with the 10MHz bus speed being termed *fast*. Subsequent enhancements pushed bus speeds to *ultra* (20MHz), *fast-40* (40MHz), and *fast-80*[5]. Further increases in transfer rates lead to several different versions of the *ultra160* bus speed.

By combining these terms, various SCSI configurations can be concisely named. For example, "ultra-wide SCSI" refers to a 16-bit SCSI bus running at 20MHz.

The original SCSI standard used *single-ended* signaling; this is an electrical configuration where only one conductor is used to pass an electrical signal. Later implementations also permitted the use of *differential* signaling, where two conductors are used to pass a signal. Differential SCSI (which was later renamed to *high voltage differential* or HVD SCSI) had the benefit of reduced sensitivity to electrical noise and allowed longer cable lengths, but it never became popular in the mainstream computer market. A later implementation, known as *low voltage differential* (LVD), has finally broken through to the mainstream and is a requirement for the higher bus speeds.

The width of a SCSI bus not only dictates the amount of data that can be transferred with each clock cycle, but it also determines how many devices can be connected to a bus. Regular SCSI supports 8 uniquely-addressed devices, while wide SCSI supports 16. In either case, you must make sure that all devices are set to use a unique SCSI ID. Two devices sharing a single ID causes problems that could lead to data corruption.

One other thing to keep in mind is that *every* device on the bus uses an ID. *This includes the SCSI controller.* Quite often system administrators forget this and unwittingly set a device to use the same SCSI ID as the bus's controller. This also means that, in practice, only 7 (or 15, for wide SCSI) devices may be present on a single bus, as each bus must reserve an ID for the controller.

[4] Some storage hardware (usually those that incorporate removable drive "carriers") is designed so that the act of plugging a module into place automatically sets the SCSI ID to an appropriate value.

[5] Fast-80 is not technically a change in bus speed; instead the 40MHz bus was retained, but data was clocked at both the rising and falling of each clock pulse, effectively doubling the throughput.

Note

Most SCSI implementations include some means of scanning the SCSI bus; this is often used to confirm that all the devices are properly configured. If a bus scan returns the same device for every single SCSI ID, that device has been incorrectly set to the same SCSI ID as the SCSI controller. To resolve the problem, reconfigure the device to use a different (and unique) SCSI ID.

Because of SCSI's bus-oriented architecture, it is necessary to properly *terminate* both ends of the bus. Termination is accomplished by placing a load of the correct electrical impedance on each conductor comprising the SCSI bus. Termination is an electrical requirement; without it, the various signals present on the bus would be reflected off the ends of the bus, garbling all communication.

Many (but not all) SCSI devices come with internal terminators that can be enabled or disabled using jumpers or switches. External terminators are also available.

One last thing to keep in mind about SCSI -- it is not just an interface standard for mass storage devices. Many other devices (such as scanners, printers, and communications devices) use SCSI. Although these are much less common than SCSI mass storage devices, they do exist. However, it is likely that, with the advent of USB and IEEE-1394 (often called Firewire), these interfaces will be used more for these types of devices in the future.

Note

The USB and IEEE-1394 interfaces are also starting to make inroads in the mass storage arena; however, no native USB or IEEE-1394 mass-storage devices currently exist. Instead, the present-day offerings are based on ATA or SCSI devices with external conversion circuitry.

No matter what interface a mass storage device uses, the inner workings of the device has a bearing on its performance. The following section explores this important subject.

5.4. Hard Drive Performance Characteristics

Hard drive performance characteristics have already been introduced in *Section 4.2.4, “Hard Drives”*; this section discusses the matter in more depth. This is important for system administrators to understand, because without at least basic knowledge of how hard drives operate, it is possible to unwittingly making changes to your system configuration that could negatively impact its performance.

The time it takes for a hard drive to respond to and complete an I/O request is dependent on two things:

- The hard drive's mechanical and electrical limitations
- The I/O load imposed by the system

The following sections explore these aspects of hard drive performance in more depth.

5.4.1. Mechanical/Electrical Limitations

Because hard drives are electro-mechanical devices, they are subject to various limitations on their speed and performance. Every I/O request requires the various components of the drive to work together to satisfy the request. Because each of these components have different performance characteristics, the overall performance of the hard drive is determined by the sum of the performance of the individual components.

However, the electronic components are at least an order of magnitude faster than the mechanical components. Therefore, it is the mechanical components that have the greatest impact on overall hard drive performance.

Note

The most effective way to improve hard drive performance is to reduce the drive's mechanical activity as much as possible.

The average access time of a typical hard drive is roughly 8.5 milliseconds. The following sections break this figure down in more detail, showing how each component impacts the hard drive's overall performance.

5.4.1.1. Command Processing Time

All hard drives produced today have sophisticated embedded computer systems controlling their operation. These computer systems perform the following tasks:

- Interacting with the outside world via hard drive's interface
- Controlling the operation of the rest of the hard drive's components, recovering from any error conditions that might arise
- Processing the raw data read from and written to the actual storage media

Even though the microprocessors used in hard drives are relatively powerful, the tasks assigned to them take time to perform. On average, this time is in the range of .003 milliseconds.

5.4.1.2. Heads Reading/Writing Data

The hard drive's read/write heads only work when the disk platters over which they "fly" are spinning. Because it is the movement of the media under the heads that allows the data to be read or written, the time that it takes for media containing the desired sector to pass completely underneath the head is the sole determinant of the head's contribution to total access time. This averages .0086 milliseconds for a 10,000 RPM drive with 700 sectors per track.

5.4.1.3. Rotational Latency

Because a hard drive's disk platters are continuously spinning, when the I/O request arrives it is highly unlikely that the platter will be at exactly the right point in its rotation necessary to access the desired sector. Therefore, even if the rest of the drive is ready to access that sector, it is necessary for everything to wait while the platter rotates, bringing the desired sector into position under the read/write head.

This is the reason why higher-performance hard drives typically rotate their disk platters at higher speeds. Today, speeds of 15,000 RPM are reserved for the highest-performing drives, while 5,400 RPM is considered adequate only for entry-level drives. This averages approximately 3 milliseconds for a 10,000 RPM drive.

5.4.1.4. Access Arm Movement

If there is one component in hard drives that can be considered its Achilles' Heel, it is the access arm. The reason for this is that the access arm must move very quickly and accurately over relatively long distances. In addition, the access arm movement is not continuous -- it must rapidly accelerate as it approaches the desired cylinder and then just as rapidly decelerate to avoid overshooting. Therefore, the access arm must be strong (to survive the violent forces caused by the need for quick movement) but also light (so that there is less mass to accelerate/decelerate).

Achieving these conflicting goals is difficult, a fact that is shown by how relatively much time the access arm movement takes when compared to the time taken by the other components. Therefore, the movement of the access arm is the primary determinant of a hard drive's overall performance, averaging 5.5 milliseconds.

5.4.2. I/O Loads and Performance

The other thing that controls hard drive performance is the I/O load to which a hard drive is subjected. Some of the specific aspects of the I/O load are:

- The amount of reads versus writes
- The number of current readers/writers
- The locality of reads/writes

These are discussed in more detail in the following sections.

5.4.2.1. Reads Versus Writes

For the average hard drive using magnetic media for data storage, the number of read I/O operations versus the number of write I/O operations is not of much concern, as reading and writing data take the same amount of time[6]. However, other mass storage technologies take different amounts of time to process reads and writes[7].

The impact of this is that devices that take longer to process write I/O operations (for example) are able to handle fewer write I/Os than read I/Os. Looked at another way, a write I/O consumes more of the device's ability to process I/O requests than does a read I/O.

5.4.2.2. Multiple Readers/Writers

A hard drive that processes I/O requests from multiple sources experiences a different load than a hard drive that services I/O requests from only one source. The main reason for this is due to the fact that multiple I/O requesters have the potential to bring higher I/O loads to bear on a hard drive than a single I/O requester.

This is because the I/O requester must perform some amount of processing before an I/O can take place. After all, the requester must determine the nature of the I/O request before it can be performed. Because the processing necessary to make this determination takes time, there is an upper limit

[6] Actually, this is not entirely true. All hard drives include some amount of on-board cache memory that is used to improve read performance. However, any I/O request to read data must eventually be satisfied by physically reading the data from the storage medium. This means that, while cache may alleviate read I/O performance problems, it can never totally eliminate the time required to physically read the data from the storage medium.

[7] Some optical disk drives exhibit this behavior, due to the physical constraints of the technologies used to implement optical data storage.

on the I/O load that any one requester can generate -- only a faster CPU can raise it. This limitation becomes more pronounced if the requester requires human input before performing an I/O.

However, with multiple requesters, higher I/O loads may be sustained. As long as sufficient CPU power is available to support the processing necessary to generate the I/O requests, adding more I/O requesters increases the resulting I/O load.

However, there is another aspect to this that also has a bearing on the resulting I/O load. This is discussed in the following section.

5.4.2.3. Locality of Reads/Writes

Although not strictly constrained to a multi-requester environment, this aspect of hard drive performance does tend to show itself more in such an environment. The issue is whether the I/O requests being made of a hard drive are for data that is physically close to other data that is also being requested.

The reason why this is important becomes apparent if the electromechanical nature of the hard drive is kept in mind. The slowest component of any hard drive is the access arm. Therefore, if the data being accessed by the incoming I/O requests requires no movement of the access arm, the hard drive is able to service many more I/O requests than if the data being accessed was spread over the entire drive, requiring extensive access arm movement.

This can be illustrated by looking at hard drive performance specifications. These specifications often include *adjacent cylinder seek times* (where the access arm is moved a small amount -- only to the next cylinder), and *full-stroke seek times* (where the access arm moves from the very first cylinder to the very last one). For example, here are the seek times for a high-performance hard drive:

Table 5.4. Adjacent Cylinder and Full-Stroke Seek Times (in Milliseconds)

Adjacent Cylinder	Full-Stroke
0.6	8.2

5.5. Making the Storage Usable

Once a mass storage device is in place, there is little that it can be used for. True, data can be written to it and read back from it, but without any underlying structure data access is only possible by using sector addresses (either geometrical or logical).

What is needed are methods of making the raw storage a hard drive provides more easily usable. The following sections explore some commonly-used techniques for doing just that.

5.5.1. Partitions/Slices

The first thing that often strikes a system administrator is that the size of a hard drive may be much larger than necessary for the task at hand. As a result, many operating systems have the capability of dividing a hard drive's space into various *partitions* or *slices*.

Because they are separate from each other, partitions can have different amounts of space utilized, and that space in no way impacts the space utilized by other partitions. For example, the partition holding the files comprising the operating system is not affected even if the partition holding the users' files becomes full. The operating system still has free space for its own use.

Although it is somewhat simplistic, you can think of partitions as being similar to individual disk drives. In fact, some operating systems actually refer to partitions as "drives". However, this viewpoint is not entirely accurate; therefore, it is important that we look at partitions more closely.

5.5.1.1. Partition Attributes

Partitions are defined by the following attributes:

- Partition geometry
- Partition type
- Partition type field

These attributes are explored in more detail in the following sections.

5.5.1.1.1. Geometry

A partition's geometry refers to its physical placement on a disk drive. The geometry can be specified in terms of starting and ending cylinders, heads, and sectors, although most often partitions start and end on cylinder boundaries. A partition's size is then defined as the amount of storage between the starting and ending cylinders.

5.5.1.1.2. Partition Type

The partition type refers to the partition's relationship with the other partitions on the disk drive. There are three different partition types:

- Primary partitions
- Extended partitions
- Logical partitions

The following sections describe each partition type.

5.5.1.1.2.1. Primary Partitions

Primary partitions are partitions that take up one of the four primary partition slots in the disk drive's partition table.

5.5.1.1.2.2. Extended Partitions

Extended partitions were developed in response to the need for more than four partitions per disk drive. An extended partition can itself contain multiple partitions, greatly extending the number of partitions possible on a single drive. The introduction of extended partitions was driven by the ever-increasing capacities of new disk drives.

5.5.1.1.2.3. Logical Partitions

Logical partitions are those partitions contained within an extended partition; in terms of use they are no different than a non-extended primary partition.

5.5.1.1.3. Partition Type Field

Each partition has a type field that contains a code indicating the partition's anticipated usage. The type field may or may not reflect the computer's operating system. Instead, it may reflect how data is to be stored within the partition. The following section contains more information on this important point.

5.5.2. File Systems

Even with the proper mass storage device, properly configured, and appropriately partitioned, we would still be unable to store and retrieve information easily -- we are missing a way of structuring and organizing that information. What we need is a *file system*.

The concept of a file system is so fundamental to the use of mass storage devices that the average computer user often does not even make the distinction between the two. However, system administrators cannot afford to ignore file systems and their impact on day-to-day work.

A file system is a method of representing data on a mass storage device. File systems usually include the following features:

- File-based data storage
- Hierarchical directory (sometimes known as "folder") structure
- Tracking of file creation, access, and modification times
- Some level of control over the type of access allowed for a specific file
- Some concept of file ownership
- Accounting of space utilized

Not all file systems posses every one of these features. For example, a file system constructed for a single-user operating system could easily use a more simplified method of access control and could conceivably do away with support for file ownership altogether.

One point to keep in mind is that the file system used can have a large impact on the nature of your daily workload. By ensuring that the file system you use in your organization closely matches your organization's functional requirements, you can ensure that not only is the file system up to the task, but that it is more easily and efficiently maintainable.

With this in mind, the following sections explore these features in more detail.

5.5.2.1. File-Based Storage

While file systems that use the file metaphor for data storage are so nearly universal as to be considered a given, there are still some aspects that should be considered here.

First is to be aware of any restrictions on file names. For instance, what characters are permitted in a file name? What is the maximum file name length? These questions are important, as it dictates those file names that can be used and those that cannot. Older operating systems with more primitive file systems often allowed only alphanumeric characters (and only uppercase at that), and only traditional *8.3* file names (meaning an eight-character file name, followed by a three-character file extension).

5.5.2.2. Hierarchical Directory Structure

While the file systems used in some very old operating systems did not include the concept of directories, all commonly-used file systems today include this feature. Directories are themselves usually implemented as files, meaning that no special utilities are required to maintain them.

Furthermore, because directories are themselves files, and directories contain files, directories can therefore contain other directories, making a multi-level directory hierarchy possible. This is a powerful concept with which all system administrators should be thoroughly familiar. Using multi-level directory hierarchies can make file management much easer for you and for your users.

5.5.2.3. Tracking of File Creation, Access, Modification Times

Most file systems keep track of the time at which a file was created; some also track modification and access times. Over and above the convenience of being able to determine when a given file was created, accessed, or modified, these dates are vital for the proper operation of incremental backups.

More information on how backups make use of these file system features can be found in *Section 8.2, “Backups”*.

5.5.2.4. Access Control

Access control is one area where file systems differ dramatically. Some file systems have no clear-cut access control model, while others are much more sophisticated. In general terms, most modern day file systems combine two components into a cohesive access control methodology:

- User identification
- Permitted action list

User identification means that the file system (and the underlying operating system) must first be capable of uniquely identifying individual users. This makes it possible to have full accountability with respect to any operations on the file system level. Another often-helpful feature is that of user *groups* -- creating ad-hoc collections of users. Groups are most often used by organizations where users may be members of one or more projects. Another feature that some file systems support is the creation of generic identifiers that can be assigned to one or more users.

Next, the file system must be capable of maintaining lists of actions that are permitted (or not permitted) against each file. The most commonly-tracked actions are:

- Reading the file
- Writing the file
- Executing the file

Various file systems may extend the list to include other actions such as deleting, or even the ability to make changes related to a file's access control.

5.5.2.5. Accounting of Space Utilized

One constant in a system administrator's life is that there is never enough free space, and even if there is, it will not remain free for long. Therefore, a system administrator should at least be able to easily determine the level of free space available for each file system. In addition, file systems with well-defined user identification capabilities often include the capability to display the amount of space a particular user has consumed.

This feature is vital in large multi-user environments, as it is an unfortunate fact of life that the 80/20 rule often applies to disk space -- 20 percent of your users will be responsible for consuming 80 percent of your available disk space. By making it easy to determine which users are in that 20 percent, you can more effectively manage your storage-related assets.

Taking this a step further, some file systems include the ability to set per-user limits (often known as *disk quotas*) on the amount of disk space that can be consumed. The specifics vary from file system to file system, but in general each user can be assigned a specific amount of storage that a user can use. Beyond that, various file systems differ. Some file systems permit the user to exceed their limit for one time only, while others implement a "grace period" during which a second, higher limit is applied.

5.5.3. Directory Structure

Many system administrators give little thought to how the storage they make available to users today is actually going to be used tomorrow. However, a bit of thought spent on this matter before handing over the storage to users can save a great deal of unnecessary effort later on.

The main thing that system administrators can do is to use directories and subdirectories to structure the storage available in an understandable way. There are several benefits to this approach:

- More easily understood
- More flexibility in the future

By enforcing some level of structure on your storage, it can be more easily understood. For example, consider a large mult-user system. Instead of placing all user directories in one large directory, it might make sense to use subdirectories that mirror your organization's structure. In this way, people that work in accounting have their directories under a directory named **`accounting`**, people that work in engineering would have their directories under **`engineering`**, and so on.

The benefits of such an approach are that it would be easier on a day-to-day basis to keep track of the storage needs (and usage) for each part of your organization. Obtaining a listing of the files used by everyone in human resources is straightforward. Backing up all the files used by the legal department is easy.

With the appropriate structure, flexibility is increased. To continue using the previous example, assume for a moment that the engineering department is due to take on several large new projects. Because of this, many new engineers are to be hired in the near future. However, there is currently not enough free storage available to support the expected additions to engineering.

However, since every person in engineering has their files stored under the **`engineering`** directory, it would be a straightforward process to:

- Procure the additional storage necessary to support engineering
- Back up everything under the **`engineering`** directory
- Restore the backup onto the new storage
- Rename the **`engineering`** directory on the original storage to something like **`engineering-archive`** (before deleting it entirely after running smoothly with the new configuration for a month)
- Make the necessary changes so that all engineering personnel can access their files on the new storage

Of course, such an approach does have its shortcomings. For example, if people frequently move between departments, you must have a way of being informed of such transfers, and you must modify the directory structure appropriately. Otherwise, the structure no longer reflects reality, which makes more work -- not less -- for you in the long run.

5.5.4. Enabling Storage Access

Once a mass storage device has been properly partitioned, and a file system written to it, the storage is available for general use.

For some operating systems, this is true; as soon as the operating system detects the new mass storage device, it can be formatted by the system administrator and may be accessed immediately with no additional effort.

Other operating systems require an additional step. This step -- often referred to as *mounting* -- directs the operating system as to how the storage may be accessed. Mounting storage normally is done via a special utility program or command, and requires that the mass storage device (and possibly the partition as well) be explicitly identified.

5.6. Advanced Storage Technologies

Although everything presented in this chapter so far has dealt only with single hard drives directly-attached to a system, there are other, more advanced options that you can explore. The following sections describe some of the more common approaches to expanding your mass storage options.

5.6.1. Network-Accessible Storage

Combining network and mass storage technologies can result in a great deal more flexibility for system administrators. There are two benefits that are possible with this type of configuration:

- Consolidation of storage
- Simplified administration

Storage can be consolidated by deploying high-performance servers with high-speed network connectivity and configured with large amounts of fast storage. Given an appropriate configuration, it is possible to provide storage access at speeds comparable to locally-attached storage. Furthermore, the shared nature of such a configuration often makes it possible to reduce costs, as the expenses associated with providing centralized, shared storage can be less than providing the equivalent storage for each and every client. In addition, free space is consolidated, instead of being spread out (and not widely usable) across many clients.

Centralized storage servers can also make many administrative tasks easier. For instance, monitoring free space is much easier when the storage to be monitored exists on a centralized storage server. Backups can be vastly simplified using a centralized storage server. Network-aware backups for multiple clients are possible, but require more work to configure and maintain.

There are a number of different networked storage technologies available; choosing one can be difficult. Nearly every operating system on the market today includes some means of accessing network-accessible storage, but the different technologies are incompatible with each other. What is the best approach to determining which technology to deploy?

The approach that usually provides the best results is to let the built-in capabilities of the client decide the issue. There are a number of reasons for this:

- Minimal client integration issues
- Minimal work on each client system
- Low per-client cost of entry

Keep in mind that any client-related issues are multiplied by the number of clients in your organization. By using the clients' built-in capabilities, you have no additional software to install on each client (incurring zero additional cost in software procurement). And you have the best chance for good support and integration with the client operating system.

There is a downside, however. This means that the server environment must be up to the task of providing good support for the network-accessible storage technologies required by the clients. In cases where the server and client operating systems are one and the same, there is normally no issue. Otherwise, it will be necessary to invest time and effort in making the server "speak" the clients' language. However, often this trade-off is more than justified.

5.6.2. RAID-Based Storage

One skill that a system administrator should cultivate is the ability to look at complex system configurations, and observe the different shortcomings inherent in each configuration. While this might, at first glance, seem to be a rather depressing viewpoint to take, it can be a great way to look beyond the shiny new boxes and visualize some future Saturday night with all production down due to a failure that could easily have been avoided with a bit of forethought.

With this in mind, let us use what we now know about disk-based storage and see if we can determine the ways that disk drives can cause problems. First, consider an outright hardware failure:

A disk drive with four partitions on it dies completely: what happens to the data on those partitions?

It is immediately unavailable (at least until the failing unit can be replaced, and the data restored from a recent backup).

A disk drive with a single partition on it is operating at the limits of its design due to massive I/O loads: what happens to applications that require access to the data on that partition?

The applications slow down because the disk drive cannot process reads and writes any faster.

You have a large data file that is slowly growing in size; soon it will be larger than the largest disk drive available for your system. What happens then?

The disk drive fills up, the data file stops growing, and its associated applications stop running.

Just one of these problems could cripple a data center, yet system administrators must face these kinds of issues every day. What can be done?

Fortunately, there is one technology that can address each one of these issues. The name for that technology is *RAID*.

5.6.2.1. Basic Concepts

RAID is an acronym standing for Redundant Array of Independent Disks[8]. As the name implies, RAID is a way for multiple disk drives to act as if they were a single disk drive.

RAID techniques were first developed by researchers at the University of California, Berkeley in the mid-1980s. At the time, there was a large gap in price between the high-performance disk drives used on the large computer installations of the day, and the smaller, slower disk drives used by the still-young personal computer industry. RAID was viewed as a method of having several less expensive disk drives fill in for one higher-priced unit.

More importantly, RAID arrays can be constructed in different ways, resulting in different characteristics depending on the final configuration. Let us look at the different configurations (known as RAID *levels*) in more detail.

5.6.2.1.1. RAID Levels

The Berkeley researchers originally defined five different RAID levels and numbered them "1" through "5." In time, additional RAID levels were defined by other researchers and members of the storage industry. Not all RAID levels were equally useful; some were of interest only for research purposes, and others could not be economically implemented.

[8] When early RAID research began, the acronym stood for Redundant Array of *Inexpensive* Disks, but over time the "standalone" disks that RAID was intended to supplant became cheaper and cheaper, rendering the price comparison meaningless.

In the end, there were three RAID levels that ended up seeing widespread usage:

- Level 0
- Level 1
- Level 5

The following sections discuss each of these levels in more detail.

5.6.2.1.1.1. RAID 0

The disk configuration known as RAID level 0 is a bit misleading, as this is the only RAID level that employs absolutely no redundancy. However, even though RAID 0 has no advantages from a reliability standpoint, it does have other benefits.

A RAID 0 array consists of two or more disk drives. The available storage capacity on each drive is divided into *chunks*, which represent some multiple of the drives' native block size. Data written to the array is be written, chunk by chunk, to each drive in the array. The chunks can be thought of as forming stripes across each drive in the array; hence the other term for RAID 0: *striping*.

For example, with a two-drive array and a 4KB chunk size, writing 12KB of data to the array would result in the data being written in three 4KB chunks to the following drives:

- The first 4KB would be written to the first drive, into the first chunk
- The second 4KB would be written to the second drive, into the first chunk
- The last 4KB would be written to the first drive, into the second chunk

Compared to a single disk drive, the advantages to RAID 0 include:

- Larger total size -- RAID 0 arrays can be constructed that are larger than a single disk drive, making it easier to store larger data files
- Better read/write performance -- The I/O load on a RAID 0 array is spread evenly among all the drives in the array (Assuming all the I/O is not concentrated on a single chunk)
- No wasted space -- All available storage on all drives in the array are available for data storage

Compared to a single disk drive, RAID 0 has the following disadvantage:

- Less reliability -- Every drive in a RAID 0 array must be operative for the array to be available; a single drive failure in an *N*-drive RAID 0 array results in the removal of 1/*N*th of all the data, rendering the array useless

Note

If you have trouble keeping the different RAID levels straight, just remember that RAID 0 has *zero* percent redundancy.

5.6.2.1.1.2. RAID 1

RAID 1 uses two (although some implementations support more) identical disk drives. All data is written to both drives, making them mirror images of each other. That is why RAID 1 is often known as *mirroring*.

Whenever data is written to a RAID 1 array, two physical writes must take place: one to the first drive, and one to the second drive. Reading data, on the other hand, only needs to take place once and either drive in the array can be used.

Compared to a single disk drive, a RAID 1 array has the following advantages:

- Improved redundancy -- Even if one drive in the array were to fail, the data would still be accessible
- Improved read performance -- With both drives operational, reads can be evenly split between them, reducing per-drive I/O loads

When compared to a single disk drive, a RAID 1 array has some disadvantages:

- Maximum array size is limited to the largest single drive available.
- Reduced write performance -- Because both drives must be kept up-to-date, all write I/Os must be performed by both drives, slowing the overall process of writing data to the array
- Reduced cost efficiency -- With one entire drive dedicated to redundancy, the cost of a RAID 1 array is at least double that of a single drive

Note

If you have trouble keeping the different RAID levels straight, just remember that RAID 1 has *one* hundred percent redundancy.

5.6.2.1.1.3. RAID 5

RAID 5 attempts to combine the benefits of RAID 0 and RAID 1, while minimizing their respective disadvantages.

Like RAID 0, a RAID 5 array consists of multiple disk drives, each divided into chunks. This allows a RAID 5 array to be larger than any single drive. Like a RAID 1 array, a RAID 5 array uses some disk space in a redundant fashion, improving reliability.

However, the way RAID 5 works is unlike either RAID 0 or 1.

A RAID 5 array must consist of at least three identically-sized disk drives (although more drives may be used). Each drive is divided into chunks and data is written to the chunks in order. However, not every chunk is dedicated to data storage as it is in RAID 0. Instead, in an array with *n* disk drives in it, every *n*th chunk is dedicated to *parity*.

Chunks containing parity make it possible to recover data should one of the drives in the array fail. The parity in chunk *x* is calculated by mathematically combining the data from each chunk *x* stored on all the other drives in the array. If the data in a chunk is updated, the corresponding parity chunk must be recalculated and updated as well.

This also means that every time data is written to the array, at least *two* drives are written to: the drive holding the data, and the drive containing the parity chunk.

One key point to keep in mind is that the parity chunks are not concentrated on any one drive in the array. Instead, they are spread evenly across all the drives. Even though dedicating a specific drive to contain nothing but parity is possible (in fact, this configuration is known as RAID level 4), the constant updating of parity as data is written to the array would mean that the parity drive could become a performance bottleneck. By spreading the parity information evenly throughout the array, this impact is reduced.

However, it is important to keep in mind the impact of parity on the overall storage capacity of the array. Even though the parity information is spread evenly across all the drives in the array, the amount of available storage is reduced by the size of one drive.

Compared to a single drive, a RAID 5 array has the following advantages:

- Improved redundancy -- If one drive in the array fails, the parity information can be used to reconstruct the missing data chunks, all while keeping the array available for use[9]

- Improved read performance -- Due to the RAID 0-like way data is divided between drives in the array, read I/O activity is spread evenly between all the drives

- Reasonably good cost efficiency -- For a RAID 5 array of *n* drives, only 1/*n*th of the total available storage is dedicated to redundancy

Compared to a single drive, a RAID 5 array has the following disadvantage:

- Reduced write performance -- Because each write to the array results in at least two writes to the physical drives (one write for the data and one for the parity), write performance is worse than a single drive[10]

5.6.2.1.1.4. Nested RAID Levels

As should be obvious from the discussion of the various RAID levels, each level has specific strengths and weaknesses. It was not long after RAID-based storage began to be deployed that people began to wonder whether different RAID levels could somehow be combined, producing arrays with all of the strengths and none of the weaknesses of the original levels.

For example, what if the disk drives in a RAID 0 array were themselves actually RAID 1 arrays? This would give the advantages of RAID 0's speed, with the reliability of RAID 1.

This is just the kind of thing that can be done. Here are the most commonly-nested RAID levels:

- RAID 1+0

- RAID 5+0

- RAID 5+1

Because nested RAID is used in more specialized environments, we will not go into greater detail here. However, there are two points to keep in mind when thinking about nested RAID:

- Order matters -- The order in which RAID levels are nested can have a large impact on reliability. In other words, RAID 1+0 and RAID 0+1 are *not* the same.

- Costs can be high -- If there is any disadvantage common to all nested RAID implementations, it is one of cost; for example, the smallest possible RAID 5+1 array consists of six disk drives (and even more drives are required for larger arrays).

Now that we have explored the concepts behind RAID, let us see how RAID can be implemented.

5.6.2.1.2. RAID Implementations

It is obvious from the previous sections that RAID requires additional "intelligence" over and above the usual disk I/O processing for individual drives. At the very least, the following tasks must be performed:

- Dividing incoming I/O requests to the individual disks in the array

- For RAID 5, calculating parity and writing it to the appropriate drive in the array
- Monitoring the individual disks in the array and taking the appropriate action should one fail
- Controlling the rebuilding of an individual disk in the array, when that disk has been replaced or repaired
- Providing a means to allow administrators to maintain the array (removing and adding drives, initiating and halting rebuilds, etc.)

There are two major methods that may be used to accomplish these tasks. The next two sections describe them in more detail.

5.6.2.1.2.1. Hardware RAID

A hardware RAID implementation usually takes the form of a specialized disk controller card. The card performs all RAID-related functions and directly controls the individual drives in the arrays attached to it. With the proper driver, the arrays managed by a hardware RAID card appear to the host operating system just as if they were regular disk drives.

Most RAID controller cards work with SCSI drives, although there are some ATA-based RAID controllers as well. In any case, the administrative interface is usually implemented in one of three ways:

- Specialized utility programs that run as applications under the host operating system, presenting a software interface to the controller card
- An on-board interface using a serial port that is accessed using a terminal emulator
- A BIOS-like interface that is only accessible during the system's power-up testing

Some RAID controllers have more than one type of administrative interface available. For obvious reasons, a software interface provides the most flexibility, as it allows administrative functions while the operating system is running. However, if you are booting an operating system from a RAID controller, an interface that does not require a running operating system is a requirement.

Because there are so many different RAID controller cards on the market, it is impossible to go into further detail here. The best course of action is to read the manufacturer's documentation for more information.

5.6.2.1.2.2. Software RAID

Software RAID is RAID implemented as kernel- or driver-level software for a particular operating system. As such, it provides more flexibility in terms of hardware support -- as long as the hardware is supported by the operating system, RAID arrays can be configured and deployed. This can dramatically reduce the cost of deploying RAID by eliminating the need for expensive, specialized RAID hardware.

Often the excess CPU power available for software RAID parity calculations greatly exceeds the processing power present on a RAID controller card. Therefore, some software RAID implementations actually have the capability for higher performance than hardware RAID implementations.

However, software RAID does have limitations not present in hardware RAID. The most important one to consider is support for booting from a software RAID array. In most cases, only RAID 1 arrays can be used for booting, as the computer's BIOS is not RAID-aware. Since a single drive from a RAID 1 array is indistinguishable from a non-RAID boot device, the BIOS can successfully start the boot process; the operating system can then change over to software RAID operation once it has gained control of the system.

5.6.3. Logical Volume Management

One other advanced storage technology is that of *logical volume management* (LVM). LVM makes it possible to treat physical mass storage devices as low-level building blocks on which different storage configurations are built. The exact capabilities vary according to the specific implementation, but can include physical storage grouping, logical volume resizing, and data migration.

5.6.3.1. Physical Storage Grouping

Although the name given to this capability may differ, physical storage grouping is the foundation for all LVM implementations. As the name implies, the physical mass storage devices can be grouped together in such a way as to create one or more logical mass storage devices. The logical mass storage devices (or logical volumes) can be larger in capacity than the capacity of any one of the underlying physical mass storage devices.

For example, given two 100GB drives, a 200GB logical volume can be created. However, a 150GB and a 50GB logical volume could also be created. Any combination of logical volumes equal to or less than the total capacity (200GB in this example) is possible. The choices are limited only by your organization's needs.

This makes it possible for a system administrator to treat all storage as being part of a single pool, available for use in any amount. In addition, drives can be added to the pool at a later time, making it a straightforward process to stay ahead of your users' demand for storage.

5.6.3.2. Logical Volume Resizing

The feature that most system administrators appreciate about LVM is its ability to easily direct storage where it is needed. In a non-LVM system configuration, running out of space means -- at best -- moving files from the full device to one with available space. Often it can mean actual reconfiguration of your system's mass storage devices; a task that would have to take place after normal business hours.

However, LVM makes it possible to easily increase the size of a logical volume. Assume for a moment that our 200GB storage pool was used to create a 150GB logical volume, with the remaining 50GB held in reserve. If the 150GB logical volume became full, LVM makes it possible to increase its size (say, by 10GB) without any physical reconfiguration. Depending on the operating system environment, it may be possible to do this dynamically or it might require a short amount of downtime to actually perform the resizing.

5.6.3.3. Data Migration

Most seasoned system administrators would be impressed by LVM capabilities so far, but they would also be asking themselves this question:

What happens if one of the drives making up a logical volume starts to fail?

The good news is that most LVM implementations include the ability to *migrate* data off of a particular physical drive. For this to work, there must be sufficient reserve capacity left to absorb the loss of the failing drive. Once the migration is complete, the failing drive can then be replaced and added back into the available storage pool.

5.6.3.4. With LVM, Why Use RAID?

Given that LVM has some features similar to RAID (the ability to dynamically replace failing drives, for instance), and some features providing capabilities that cannot be matched by most RAID implementations (such as the ability to dynamically add more storage to a central storage pool), many people wonder whether RAID is no longer important.

Nothing could be further from the truth. RAID and LVM are complementary technologies that can be used together (in a manner similar to nested RAID levels), making it possible to get the best of both worlds.

5.7. Storage Management Day-to-Day

System administrators must pay attention to storage in the course of their day-to-day routine. There are various issues that should be kept in mind:

- Monitoring free space
- Disk quota issues
- File-related issues
- Directory-related issues
- Backup-related issues
- Performance-related issues
- Adding/removing storage

The following sections discuss each of these issues in more detail.

5.7.1. Monitoring Free Space

Making sure there is sufficient free space available should be at the top of every system administrator's daily task list. The reason why regular, frequent free space checking is so important is because free space is so dynamic; there can be more than enough space one moment, and almost none the next.

In general, there are three reasons for insufficient free space:

- Excessive usage by a user
- Excessive usage by an application
- Normal growth in usage

These reasons are explored in more detail in the following sections.

5.7.1.1. Excessive Usage by a User

Different people have different levels of neatness. Some people would be horrified to see a speck of dust on a table, while others would not think twice about having a collection of last year's pizza boxes stacked by the sofa. It is the same with storage:

- Some people are very frugal in their storage usage and never leave any unneeded files hanging around.
- Some people never seem to find the time to get rid of files that are no longer needed.

Many times where a user is responsible for using large amounts of storage, it is the second type of person that is found to be responsible.

5.7.1.1.1. Handling a User's Excessive Usage

This is one area in which a system administrator needs to summon all the diplomacy and social skills they can muster. Quite often discussions over disk space become emotional, as people view enforcement of disk usage restrictions as making their job more difficult (or impossible), that the restrictions are unreasonably small, or that they just do not have the time to clean up their files.

The best system administrators take many factors into account in such a situation. Are the restrictions equitable and reasonable for the type of work being done by this person? Does the person seem to be using their disk space appropriately? Can you help the person reduce their disk usage in some way (by creating a backup CD-ROM of all emails over one year old, for example)? Your job during the conversation is to attempt to discover if this is, in fact, the case while making sure that someone that has no real need for that much storage cleans up their act.

In any case, the thing to do is to keep the conversation on a professional, factual level. Try to address the user's issues in a polite manner ("I understand you are very busy, but everyone else in your department has the same responsibility to not waste storage, and their average utilization is less than half of yours.") while moving the conversation toward the matter at hand. Be sure to offer assistance if a lack of knowledge/experience seems to be the problem.

Approaching the situation in a sensitive but firm manner is often better than using your authority as system administrator to force a certain outcome. For example, you might find that sometimes a compromise between you and the user is necessary. This compromise can take one of three forms:

- Provide temporary space
- Make archival backups
- Give up

You might find that the user can reduce their usage if they have some amount of temporary space that they can use without restriction. People that often take advantage of this situation find that it allows them to work without worrying about space until they get to a logical stopping point, at which time they can perform some housekeeping, and determine what files in temporary storage are really needed or not.

Warning

If you offer this situation to a user, do *not* fall into the trap of allowing this temporary space to become permanent space. Make it very clear that the space being offered is temporary, and that no guarantees can be made as to data retention; no backups of any data in temporary space are ever made.

In fact, many administrators often underscore this fact by automatically deleting any files in temporary storage that are older than a certain age (a week, for example).

Other times, the user may have many files that are so obviously old that it is unlikely continuous access to them is needed. Make sure you determine that this is, in fact, the case. Sometimes individual users are responsible for maintaining an archive of old data; in these instances, you should make a point of assisting them in that task by providing multiple backups that are treated no differently from your data center's archival backups.

However, there are times when the data is of dubious value. In these instances you might find it best to offer to make a special backup for them. You then back up the old data, and give the user the backup media, explaining that they are responsible for its safekeeping, and if they ever need access to any of the data, to ask you (or your organization's operations staff -- whatever is appropriate for your organization) to restore it.

There are a few things to keep in mind so that this does not backfire on you. First and foremost is to not include files that are likely to need restoring; do not select files that are *too* new. Next, make sure that you are able to perform a restoration if one ever is requested. This means that the backup media should be of a type that you are reasonably sure will be used in your data center for the foreseeable future.

Note

Your choice of backup media should also take into consideration those technologies that can enable the user to handle data restoration themselves. For example, even though backing up several gigabytes onto CD-R media is more work than issuing a single command and spinning it off to a 20GB tape cartridge, consider that the user can then be able to access the data on CD-R whenever they want -- without ever involving you.

5.7.1.2. Excessive Usage by an Application

Sometimes an application is responsible for excessive usage. The reasons for this can vary, but can include:

- Enhancements in the application's functionality require more storage
- An increase in the number of users using the application
- The application fails to clean up after itself, leaving no-longer-needed temporary files on disk
- The application is broken, and the bug is causing it to use more storage than it should

Your task is to determine which of the reasons from this list apply to your situation. Being aware of the status of the applications used in your data center should help you eliminate several of these reasons, as should your awareness of your users' processing habits. What remains to be done is often a bit of detective work into where the storage has gone. This should narrow down the field substantially.

At this point you must then take the appropriate steps, be it the addition of storage to support an increasingly-popular application, contacting the application's developers to discuss its file handling characteristics, or writing scripts to clean up after the application.

5.7.1.3. Normal Growth in Usage

Most organizations experience some level of growth over the long term. Because of this, it is normal to expect storage utilization to increase at a similar pace. In nearly all circumstances, ongoing monitoring can reveal the average rate of storage utilization at your organization; this rate can then be used to determine the time at which additional storage should be procured before your free space actually runs out.

If you are in the position of unexpectedly running out of free space due to normal growth, you have not been doing your job.

However, sometimes large additional demands on your systems' storage can come up unexpectedly. Your organization may have merged with another, necessitating rapid changes in the IT infrastructure (and therefore, storage). A new high-priority project may have literally sprung up overnight. Changes to an existing application may have resulted in greatly increased storage needs.

No matter what the reason, there are times when you will be taken by surprise. To plan for these instances, try to configure your storage architecture for maximum flexibility. Keeping spare storage on-hand (if possible) can alleviate the impact of such unplanned events.

5.7.2. Disk Quota Issues

Many times the first thing most people think of when they think about disk quotas is using it to force users to keep their directories clean. While there are sites where this may be the case, it also helps to look at the problem of disk space usage from another perspective. What about applications that, for one reason or another, consume too much disk space? It is not unheard of for applications to fail in ways that cause them to consume all available disk space. In these cases, disk quotas can help limit the damage caused by such errant applications, forcing it to stop *before* no free space is left on the disk.

The hardest part of implementing and managing disk quotas revolves around the limits themselves. What should they be? A simplistic approach would be to divide the disk space by the number of users and/or groups using it, and use the resulting number as the per-user quota. For example, if the system has a 100GB disk drive and 20 users, each user should be given a disk quota of no more than 5GB. That way, each user would be guaranteed 5GB (although the disk would be 100% full at that point).

For those operating systems that support it, temporary quotas could be set somewhat higher -- say 7.5GB, with a permanent quota remaining at 5GB. This would have the benefit of allowing users to permanently consume no more than their percentage of the disk, but still permitting some flexibility when a user reaches (and exceeds) their limit. When using disk quotas in this manner, you are actually over-committing the available disk space. The temporary quota is 7.5GB. If all 20 users exceeded their permanent quota at the same time and attempted to approach their temporary quota, that 100GB disk would actually have to be 150GB to allow everyone to reach their temporary quota at the same time.

However, in practice not everyone exceeds their permanent quota at the same time, making some amount of overcommitment a reasonable approach. Of course, the selection of permanent and temporary quotas is up to the system administrator, as each site and user community is different.

5.7.3. File-Related Issues

System administrators often have to deal with file-related issues. The issues include:

- File Access
- File Sharing

5.7.3.1. File Access

Issues relating to file access typically revolve around one scenario -- a user is not able to access a file they feel they should be able to access.

Often this is a case of user #1 wanting to give a copy of a file to user #2. In most organizations, the ability for one user to access another user's files is strictly curtailed, leading to this problem.

There are three approaches that could conceivably be taken:

- User #1 makes the necessary changes to allow user #2 to access the file wherever it currently exists.
- A file exchange area is created for such purposes; user #1 places a copy of the file there, which can then be copied by user #2.

- User #1 uses email to give user #2 a copy of the file.

There is a problem with the first approach -- depending on how access is granted, user #2 may have full access to all of user #1's files. Worse, it might have been done in such a way as to permit *all* users in your organization access to user #1's files. Still worse, this change may not be reversed after user #2 no longer requires access, leaving user #1's files permanently accessible by others. Unfortunately, when users are in charge of this type of situation, security is rarely their highest priority.

The second approach eliminates the problem of making all of user #1's files accessible to others. However, once the file is in the file exchange area the file is readable (and depending on the permissions, even writable) by all other users. This approach also raises the possibility of the file exchange area becoming filled with files, as users often forget to clean up after themselves.

The third approach, while seemingly an awkward solution, may actually be the preferable one in most cases. With the advent of industry-standard email attachment protocols and more intelligent email programs, sending all kinds of files via email is a mostly foolproof operation, requiring no system administrator involvement. Of course, there is the chance that a user will attempt to email a 1GB database file to all 150 people in the finance department, so some amount of user education (and possibly limitations on email attachment size) would be prudent. Still, none of these approaches deal with the situation of two or more users needing ongoing access to a single file. In these cases, other methods are required.

5.7.3.2. File Sharing

When multiple users need to share a single copy of a file, allowing access by making changes to file permissions is not the best approach. It is far preferable to formalize the file's shared status. There are several reasons for this:

- Files shared out of a user's directory are vulnerable to disappearing unexpectedly when the user either leaves the organization or does nothing more unusual than rearranging their files.

- Maintaining shared access for more than one or two additional users becomes difficult, leading to the longer-term problem of unnecessary work required whenever the sharing users change responsibilities.

Therefore, the preferred approach is to:

- Have the original user relinquish direct ownership of the file

- Create a group that will own the file

- Place the file in a shared directory that is owned by the group

- Make all users needing access to the file part of the group

Of course, this approach would work equally well with multiple files as it would with single files, and can be used to implement shared storage for large, complex projects.

5.7.4. Adding/Removing Storage

Because the need for additional disk space is never-ending, a system administrator often needs to add disk space, while sometimes also removing older, smaller drives. This section provides an overview of the basic process of adding and removing storage.

Note

On many operating systems, mass storage devices are named according to their physical connection to the system. Therefore, adding or removing mass storage devices can result in unexpected changes to device names. When adding or removing storage, always make sure you review (and update, if necessary) all device name references used by your operating system.

5.7.4.1. Adding Storage

The process of adding storage to a computer system is relatively straightforward. Here are the basic steps:

1. Installing the hardware
2. Partitioning
3. Formatting the partition(s)
4. Updating system configuration
5. Modifying backup schedule

The following sections look at each step in more detail.

5.7.4.1.1. Installing the Hardware

Before anything else can be done, the new disk drive has to be in place and accessible. While there are many different hardware configurations possible, the following sections go through the two most common situations -- adding an ATA or SCSI disk drive. Even with other configurations, the basic steps outlined here still apply.

Note

No matter what storage hardware you use, you should always consider the load a new disk drive adds to your computer's I/O subsystem. In general, you should try to spread the disk I/O load over all available channels/buses. From a performance standpoint, this is far better than putting all disk drives on one channel and leaving another one empty and idle.

5.7.4.1.1.1. Adding ATA Disk Drives

ATA disk drives are mostly used in desktop and lower-end server systems. Nearly all systems in these classes have built-in ATA controllers with multiple ATA channels -- normally two or four.

Each channel can support two devices -- one master, and one slave. The two devices are connected to the channel with a single cable. Therefore, the first step is to see which channels have available space for an additional disk drive. One of three situations is possible:

- There is a channel with only one disk drive connected to it
- There is a channel with no disk drive connected to it
- There is no space available

The first situation is usually the easiest, as it is very likely that the cable already in place has an unused connector into which the new disk drive can be plugged. However, if the cable in place only has two connectors (one for the channel and one for the already-installed disk drive), then it is necessary to replace the existing cable with a three-connector model.

Before installing the new disk drive, make sure that the two disk drives sharing the channel are appropriately configured (one as master and one as slave).

The second situation is a bit more difficult, if only for the reason that a cable must be procured so that it can connect a disk drive to the channel. The new disk drive may be configured as master or slave (although traditionally the first disk drive on a channel is normally configured as master).

In the third situation, there is no space left for an additional disk drive. You must then make a decision. Do you:

- Acquire an ATA controller card, and install it
- Replace one of the installed disk drives with the newer, larger one

Adding a controller card entails checking hardware compatibility, physical capacity, and software compatibility. Basically, the card must be compatible with your computer's bus slots, there must be an open slot for it, and it must be supported by your operating system. Replacing an installed disk drive presents a unique problem: what to do with the data on the disk? There are a few possible approaches:

- Write the data to a backup device and restore it after installing the new disk drive
- Use your network to copy the data to another system with sufficient free space, restoring the data after installing the new disk drive
- Use the space physically occupied by a third disk drive by:
  1. Temporarily removing the third disk drive
  2. Temporarily installing the new disk drive in its place
  3. Copying the data to the new disk drive
  4. Removing the old disk drive
  5. Replacing it with the new disk drive
  6. Reinstalling the temporarily-removed third disk drive
- Temporarily install the original disk drive and the new disk drive in another computer, copy the data to the new disk drive, and then install the new disk drive in the original computer

As you can see, sometimes a bit of effort must be expended to get the data (and the new hardware) where it needs to go.

5.7.4.1.1.2. Adding SCSI Disk Drives

SCSI disk drives normally are used in higher-end workstations and server systems. Unlike ATA-based systems, SCSI systems may or may not have built-in SCSI controllers; some do, while others use a separate SCSI controller card.

The capabilities of SCSI controllers (whether built-in or not) also vary widely. It may supply a narrow or wide SCSI bus. The bus speed may be normal, fast, ultra, utra2, or ultra160.

If these terms are unfamiliar to you (they were discussed briefly in *Section 5.3.2.2, “SCSI”*), you must determine the capabilities of your hardware configuration and select an appropriate new disk drive.

The best resource for this information would be the documentation for your system and/or SCSI adapter.

You must then determine how many SCSI buses are available on your system, and which ones have available space for a new disk drive. The number of devices supported by a SCSI bus varies according to the bus width:

- Narrow (8-bit) SCSI bus -- 7 devices (plus controller)
- Wide (16-bit) SCSI bus -- 15 devices (plus controller)

The first step is to see which buses have available space for an additional disk drive. One of three situations is possible:

- There is a bus with less than the maximum number of disk drives connected to it
- There is a bus with no disk drives connected to it
- There is no space available on any bus

The first situation is usually the easiest, as it is likely that the cable in place has an unused connector into which the new disk drive can be plugged. However, if the cable in place does not have an unused connector, it is necessary to replace the existing cable with one that has at least one more connector.

The second situation is a bit more difficult, if only for the reason that a cable must be procured so that it can connect a disk drive to the bus.

If there is no space left for an additional disk drive, you must make a decision. Do you:

- Acquire and install a SCSI controller card
- Replace one of the installed disk drives with the new, larger one

Adding a controller card entails checking hardware compatibility, physical capacity, and software compatibility. Basically, the card must be compatible with your computer's bus slots, there must be an open slot for it, and it must be supported by your operating system.

Replacing an installed disk drive presents a unique problem: what to do with the data on the disk? There are a few possible approaches:

- Write the data to a backup device, and restore it after installing the new disk drive
- Use your network to copy the data to another system with sufficient free space, and restore after installing the new disk drive
- Use the space physically occupied by a third disk drive by:
    1. Temporarily removing the third disk drive
    2. Temporarily installing the new disk drive in its place
    3. Copying the data to the new disk drive
    4. Removing the old disk drive
    5. Replacing it with the new disk drive
    6. Reinstalling the temporarily-removed third disk drive

- Temporarily install the original disk drive and the new disk drive in another computer, copy the data to the new disk drive, and then install the new disk drive in the original computer

Once you have an available connector in which to plug the new disk drive, you must make sure that the drive's SCSI ID is set appropriately. To do this, you must know what all of the other devices on the bus (including the controller) are using for their SCSI IDs. The easiest way to do this is to access the SCSI controller's BIOS. This is normally done by pressing a specific key sequence during the system's power-up sequence. You can then view the SCSI controller's configuration, along with the devices attached to all of its buses.

Next, you must consider proper bus termination. When adding a new disk drive, the rule is actually quite straightforward -- if the new disk drive is the last (or only) device on the bus, it must have termination enabled. Otherwise, termination must be disabled.

At this point, you can move on to the next step in the process -- partitioning your new disk drive.

5.7.4.1.2. Partitioning

Once the disk drive has been installed, it is time to create one or more partitions to make the space available to your operating system. Although the tools vary depending on the operating system, the basic steps are the same:

1. Select the new disk drive

2. View the disk drive's current partition table, to ensure that the disk drive to be partitioned is, in fact, the correct one

3. Delete any unwanted partitions that may already be present on the new disk drive

4. Create the new partition(s), being sure to specify the desired size and partition type

5. Save your changes and exit the partitioning program

Warning

When partitioning a new disk drive, it is *vital* that you are sure the disk drive you are about to partition is the correct one. Otherwise, you may inadvertently partition a disk drive that is already in use, resulting in lost data.

Also make sure you have decided on the best partition size. Always give this matter serious thought, because changing it later is much more difficult than taking a bit of time now to think things through.

5.7.4.1.3. Formatting the Partition(s)

At this point, the new disk drive has one or more partitions that have been created. However, before the space contained within those partitions can be used, the partitions must first be formatted. By formatting, you are selecting a specific file system to be used within each partition. As such, this is a pivotal time in the life of this disk drive; the choices you make now cannot be changed later without going through a great deal of work.

The actual process of formatting is done by running a utility program; the steps involved in this vary according to the operating system. Once formatting is complete, the disk drive is now properly configured for use.

Before continuing, it is always best to double-check your work by accessing the partition(s) and making sure everything is in order.

5.7.4.1.4. Updating System Configuration

If your operating system requires any configuration changes to use the new storage you have added, now is the time to make the necessary changes.

At this point you can be relatively confident that the operating system is configured properly to automatically make the new storage accessible every time the system boots (although if you can afford a quick reboot, it would not hurt to do so -- just to be sure).

The next section explores one of the most commonly-forgotten steps in the process of adding new storage.

5.7.4.1.5. Modifying the Backup Schedule

Assuming that the new storage is being used to hold data worthy of being preserved, this is the time to make the necessary changes to your backup procedures and ensure that the new storage will, in fact, be backed up. The exact nature of what you must do to make this happen depends on the way that backups are performed on your system. However, here are some points to keep in mind while making the necessary changes:

- Consider what the optimal backup frequency should be
- Determine what backup style would be most appropriate (full backups only, full with incrementals, full with differentials, etc.)
- Consider the impact of the additional storage on your backup media usage, particularly as it starts to fill up
- Judge whether the additional backup could cause the backups to take too long and start using time outside of your alloted backup window
- Make sure that these changes are communicated to the people that need to know (other system administrators, operations personnel, etc.)

Once all this is done, your new storage is ready for use.

5.7.4.2. Removing Storage

Removing disk space from a system is straightforward, with most of the steps being similar to the installation sequence (except, of course, in reverse):

1. Move any data to be saved off the disk drive
2. Modify the backup schedule so that the disk drive is no longer backed up
3. Update the system configuration
4. Erase the contents of the disk drive
5. Remove the disk drive

As you can see, compared to the installation process, there are a few extra steps to take. These steps are discussed in the following sections.

5.7.4.2.1. Moving Data Off the Disk Drive

Should there be any data on the disk drive that must be saved, the first thing to do is to determine where the data should go. This decision depends mainly on what is going to be done with the data. For example, if the data is no longer going to be actively used, it should be archived, probably in the same manner as your system backups. This means that now is the time to consider appropriate retention periods for this final backup.

> **Note**
>
> Keep in mind that, in addition to any data retention guidelines your organization may have, there may also be legal requirements for retaining data for a certain length of time. Therefore, make sure you consult with the department that had been responsible for the data while it was still in use; they should know the appropriate retention period.

On the other hand, if the data is still being used, then the data should reside on the system most appropriate for that usage. Of course, if this is the case, perhaps it would be easiest to move the data by reinstalling the disk drive on the new system. If you do this, you should make a full backup of the data before doing so -- people have dropped disk drives full of valuable data (losing everything) while doing nothing more hazardous than walking across a data center.

5.7.4.2.2. Erase the Contents of the Disk Drive

No matter whether the disk drive has valuable data or not, it is a good idea to always erase a disk drive's contents prior to reassigning or relinquishing control of it. While the obvious reason is to make sure that no sensitive information remains on the disk drive, it is also a good time to check the disk drive's health by performing a read-write test for bad blocks over the entire drive.

> **Important**
>
> Many companies (and government agencies) have specific methods of erasing data from disk drives and other data storage media. You should *always* be sure you understand and abide by these requirements; in many cases there are legal ramifications if you fail to do so. The example above should in no way be considered the ultimate method of wiping a disk drive.
>
> In addition, organizations that work with classified data may find that the final disposition of the disk drive may be subject to certain legally-mandated procedures (such as physical destruction of the drive). In these instances your organization's security department should be able to offer guidance in this matter.

5.8. A Word About Backups…

One of the most important factors when considering disk storage is that of backups. We have not covered this subject here, because an in-depth section (*Section 8.2, “Backups”*) has been dedicated to backups.

5.9. Red Hat Enterprise Linux-Specific Information

Depending on your past system administration experience, managing storage under Red Hat Enterprise Linux is either mostly familiar or completely foreign. This section discusses aspects of storage administration specific to Red Hat Enterprise Linux.

5.9.1. Device Naming Conventions

As with all Linux-like operating systems, Red Hat Enterprise Linux uses device files to access all hardware (including disk drives). However, the naming conventions for attached storage devices varies somewhat between various Linux and Linux-like implementations. Here is how these device files are named under Red Hat Enterprise Linux.

> **Note**
>
> Device names under Red Hat Enterprise Linux are determined at boot-time.
>
> Therefore, changes made to a system's hardware configuration can result in device names changing when the system reboots. Because of this, problems can result if any device name references in system configuration files are not updated appropriately.

5.9.1.1. Device Files

Under Red Hat Enterprise Linux, the device files for disk drives appear in the **`/dev/`** directory. The format for each file name depends on several aspects of the actual hardware and how it has been configured. The important points are as follows:

- Device type
- Unit
- Partition

5.9.1.1.1. Device Type

The first two letters of the device file name refer to the specific type of device. For disk drives, there are two device types that are most common:

- **sd** -- The device is SCSI-based
- **hd** -- The device is ATA-based

More information about ATA and SCSI can be found in *Section 5.3.2, “Present-Day Industry-Standard Interfaces”*.

5.9.1.1.2. Unit

Following the two-letter device type are one or two letters denoting the specific unit. The unit designator starts with "a" for the first unit, "b" for the second, and so on. Therefore, the first hard drive on your system may appear as **hda** or **sda**.

> **Note**
>
> SCSI's ability to address large numbers of devices necessitated the addition of a second unit character to support systems with more than 26 SCSI devices attached. Therefore, the first 26 SCSI hard drives on a system would be named **sda** through **`sdz`**, the next 26 would be named **`sdaa`** through **`sdaz`**, and so on.

5.9.1.1.3. Partition

The final part of the device file name is a number representing a specific partition on the device, starting with "1." The number may be one or two digits in length, depending on the number of partitions written to the specific device. Once the format for device file names is known, it is easy to understand what each refers to. Here are some examples:

- **`/dev/hda1`** -- The first partition on the first ATA drive
- **`/dev/sdb12`** -- The twelfth partition on the second SCSI drive
- **`/dev/sdad4`** -- The fourth partition on the thirtieth SCSI drive

5.9.1.1.4. Whole-Device Access

There are instances where it is necessary to access the entire device and not just a specific partition. This is normally done when the device is not partitioned or does not support standard partitions (such as a CD-ROM drive). In these cases, the partition number is omitted:

- **`/dev/hdc`** -- The entire third ATA device
- **`/dev/sdb`** -- The entire second SCSI device

However, most disk drives use partitions (more information on partitioning under Red Hat Enterprise Linux can be found in *Section 5.9.6.1, “Adding Storage”*).

5.9.1.2. Alternatives to Device File Names

Because adding or removing mass storage devices can result in changes to the device file names for existing devices, there is a risk of storage not being available when the system reboots. Here is an example of the sequence of events leading to this problem:

1. The system administrator adds a new SCSI controller so that two new SCSI drives can be added to the system (the existing SCSI bus is completely full)
2. The original SCSI drives (including the first drive on the bus: **`/dev/sda`**) are not changed in any way
3. The system is rebooted
4. The SCSI drive formerly known as **`/dev/sda`** now has a new name, because the first SCSI drive on the new controller is now **`/dev/sda`**

In theory, this sounds like a terrible problem. However, in practice it rarely is. It is rarely a problem for a number of reasons. First, hardware reconfigurations of this type happen rarely. Second, it is likely that the system administrator has scheduled downtime to make the necessary changes; downtimes require careful planning to ensure the work being done does not take longer than the alloted time. This planning has the side benefit of bringing to light any issues related to device name changes.

However, some organizations and system configurations are more likely to run into this issue. Organizations that require frequent reconfigurations of storage to meet their needs often use hardware capable of reconfiguration without requiring downtime. Such *hotpluggable* hardware makes it easy to add or remove storage. But under these circumstances device naming issues can become a problem. Fortunately, Red Hat Enterprise Linux contains features that make device name changes less of a problem.

5.9.1.2.1. File System Labels

Some file systems (which are discussed further in *Section 5.9.2, “File System Basics”*) have the ability to store a *label* -- a character string that can be used to uniquely identify the data the file system contains. Labels can then be used when mounting the file system, eliminating the need to use the device name.

File system labels work well; however, file system labels must be unique system-wide. If there is ever more than one file system with the same label, you may not be able to access the file system you intended to. Also note that system configurations which do not use file systems (some databases, for example) cannot take advantage of file system labels.

5.9.1.2.2. Using `devlabel`

The **`devlabel`** software attempts to address the device naming issue in a different manner than file system labels. The **`devlabel`** software is run by Red Hat Enterprise Linux whenever the system reboots (and whenever hotpluggable devices are inserted or removed).

When **`devlabel`** runs, it reads its configuration file (**`/etc/sysconfig/devlabel`**) to obtain the list of devices for which it is responsible. For each device on the list, there is a symbolic link (chosen by the system administrator) and the device's UUID (Universal Unique IDentifier).

The **`devlabel`** command makes sure the symbolic link always refers to the originally-specified device -- even if that device's name has changed. In this way, a system administrator can configure a system to refer to **`/dev/projdisk`** instead of **`/dev/sda12`**, for example.

Because the UUID is obtained directly from the device, **`devlabel`** must only search the system for the matching UUID and update the symbolic link appropriately.

For more information on **`devlabel`**, refer to the *System Administrators Guide*.

5.9.2. File System Basics

Red Hat Enterprise Linux includes support for many popular file systems, making it possible to easily access the file systems of other operating systems.

This is particularly useful in dual-boot scenarios and when migrating files from one operating system to another.

The supported file systems include (but are not limited to):

- EXT2
- EXT3
- NFS
- ISO 9660
- MSDOS
- VFAT

The following sections explore the more popular file systems in greater detail.

5.9.2.1. EXT2

Until recently, the ext2 file system had been the standard file system for Linux. As such, it has received extensive testing and is considered one of the more robust file systems in use today.

However, there is no perfect file system, and ext2 is no exception. One problem that is commonly reported is that an ext2 file system must undergo a lengthy file system integrity check if the system was not cleanly shut down. While this requirement is not unique to ext2, the popularity of ext2, combined with the advent of larger disk drives, meant that file system integrity checks were taking longer and longer. Something had to be done.

The next section describes the approach taken to resolve this issue under Red Hat Enterprise Linux.

5.9.2.2. EXT3

The ext3 file system builds upon ext2 by adding journaling capabilities to the already-proven ext2 codebase. As a journaling file system, ext3 always keeps the file system in a consistent state, eliminating the need for lengthy file system integrity checks.

This is accomplished by writing all file system changes to an on-disk journal, which is then flushed on a regular basis. After an unexpected system event (such as a power outage or system crash), the only operation that needs to take place prior to making the file system available is to process the contents of the journal; in most cases this takes approximately one second.

Because ext3's on-disk data format is based on ext2, it is possible to access an ext3 file system on any system capable of reading and writing an ext2 file system (without the benefit of journaling, however). This can be a sizable benefit in organizations where some systems are using ext3 and some are still using ext2.

5.9.2.3. ISO 9660

In 1987, the International Organization for Standardization (known as ISO) released standard 9660. ISO 9660 defines how files are represented on CD-ROMs. Red Hat Enterprise Linux system administrators will likely see ISO 9660-formatted data in two places:

- CD-ROMs

- Files (usually referred to as *ISO images*) containing complete ISO 9660 file systems, meant to be written to CD-R or CD-RW media

The basic ISO 9660 standard is rather limited in functionality, especially when compared with more modern file systems. File names may be a maximum of eight characters long and an extension of no more than three characters is permitted. However, various extensions to the standard have become popular over the years, among them:

- Rock Ridge -- Uses some fields undefined in ISO 9660 to provide support for features such as long mixed-case file names, symbolic links, and nested directories (in other words, directories that can themselves contain other directories)

- Joliet -- An extension of the ISO 9660 standard, developed by Microsoft to allow CD-ROMs to contain long file names, using the Unicode character set

Red Hat Enterprise Linux is able to correctly interpret ISO 9660 file systems using both the Rock Ridge and Joliet extensions.

5.9.2.4. MSDOS

Red Hat Enterprise Linux also supports file systems from other operating systems. As the name for the msdos file system implies, the original operating system supporting this file system was Microsoft's MS-DOS®. As in MS-DOS, a Red Hat Enterprise Linux system accessing an msdos file system is limited to 8.3 file names. Likewise, other file attributes such as permissions and ownership cannot be changed. However, from a file interchange standpoint, the msdos file system is more than sufficient to get the job done.

5.9.2.5. VFAT

The vfat file system was first used by Microsoft's Windows® 95 operating system. An improvement over the msdos file system, file names on a vfat file system may be longer than msdos's 8.3. However, permissions and ownership still cannot be changed.

5.9.3. Mounting File Systems

To access any file system, it is first necessary to *mount* it. By mounting a file system, you direct Red Hat Enterprise Linux to make a specific partition (on a specific device) available to the system. Likewise, when access to a particular file system is no longer desired, it is necessary to *umount* it.

To mount any file system, two pieces of information must be specified:

- A means of uniquely identifying the desired disk drive and partition, such as device file name, file system label, or **`devlabel`**-managed symbolic link

- A directory under which the mounted file system is to be made available (otherwise known as a *mount point*)

The following section discusses mount points in more detail.

5.9.3.1. Mount Points

Unless you are used to Linux (or Linux-like) operating systems, the concept of a mount point will at first seem strange. However, it is one of the most powerful and flexible methods of managing file systems developed. With many other operating systems, a full file specification includes the file name, some means of identifying the specific directory in which the file resides, and a means of identifying the physical device on which the file can be found.

With Red Hat Enterprise Linux, a slightly different approach is used. As with other operating systems, a full file specification includes the file's name and the directory in which it resides. However, there is no explicit device specifier.

The reason for this apparent shortcoming is the mount point. On other operating systems, there is one directory hierarchy for each partition. However, on Linux-like systems, there is only *one* directory hierarchy system-wide and this single hierarchy can span multiple partitions. The key is the mount point. When a file system is mounted, that file system is made available as a set of subdirectories under the specified mount point.

This apparent shortcoming is actually a strength. It means that seamless expansion of a Linux file system is possible, with every directory capable of acting as a mount point for additional disk space.

As an example, assume a Red Hat Enterprise Linux system contained a directory **`foo`** in its root directory; the full path to the directory would be **`/foo/`**. Next, assume that this system has a partition that is to be mounted, and that the partition's mount point is to be **`/foo/`**. If that partition had a file by the name of **`bar.txt`** in its top-level directory, after the partition was mounted you could access the file with the following full file specification:

```
/foo/bar.txt
```

In other words, once this partition has been mounted, any file that is read or written anywhere under the **`/foo/`** directory will be read from or written to that partition.

A commonly-used mount point on many Red Hat Enterprise Linux systems is **`/home/`** -- that is because the login directories for all user accounts are normally located under **`/home/`**. If **`/home/`** is used as a mount point, all users' files are written to a dedicated partition and will not fill up the operating system's file system.

Note

Since a mount point is just an ordinary directory, it is possible to write files into a directory that is later used as a mount point. If this happens, what happens to the files that were in the directory originally?

For as long as a partition is mounted on the directory, the files are not accessible (the mounted file system appears in place of the directory's contents). However, the files will not be harmed and can be accessed after the partition is unmounted.

5.9.3.2. Seeing What is Mounted

In addition to mounting and unmounting disk space, it is possible to see what is mounted. There are several different ways of doing this:

- Viewing **`/etc/mtab`**
- Viewing **`/proc/mounts`**
- Issuing the **`df`** command

5.9.3.2.1. Viewing `/etc/mtab`

The file **`/etc/mtab`** is a normal file that is updated by the **`mount`** program whenever file systems are mounted or unmounted. Here is a sample **`/etc/mtab`**:

```
 /dev/sda3 / ext3 rw 0 0 none /proc proc rw 0 0 usbdevfs /proc/bus/usb usbdevfs rw 0 0 /dev/
sda1 /boot ext3 rw 0 0 none /dev/pts devpts rw,gid=5,mode=620 0 0 /dev/sda4 /home ext3 rw 0 0
 none /dev/shm tmpfs rw 0 0 none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
```

Note

The **`/etc/mtab`** file is meant to be used to display the status of currently-mounted file systems only. It should not be manually modified.

Each line represents a file system that is currently mounted and contains the following fields (from left to right):

- The device specification
- The mount point
- The file system type
- Whether the file system is mounted read-only (**`ro`**) or read-write (**`rw`**), along with any other mount options
- Two unused fields with zeros in them (for compatibility with **`/etc/fstab`**[11])

5.9.3.2.2. Viewing `/proc/mounts`

The **`/proc/mounts`** file is part of the proc virtual file system. As with the other files under **`/proc/`**, the **`mounts`** "file" does not exist on any disk drive in your Red Hat Enterprise Linux system.

In fact, it is not even a file; instead it is a representation of system status made available (by the Linux kernel) in file form.

Using the command **`cat /proc/mounts`**, we can view the status of all mounted file systems:

```
 rootfs / rootfs rw 0 0 /dev/root / ext3 rw 0 0 /proc /proc proc rw 0 0 usbdevfs /proc/bus/
usb usbdevfs rw 0 0 /dev/sda1 /boot ext3 rw 0 0 none /dev/pts devpts rw 0 0 /dev/sda4 /home
 ext3 rw 0 0 none /dev/shm tmpfs rw 0 0 none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
```

As we can see from the above example, the format of **`/proc/mounts`** is very similar to that of **`/etc/mtab`**. There are a number of file systems mounted that have nothing to do with disk drives. Among these are the **`/proc/`** file system itself (along with two other file systems mounted under **`/proc/`**), pseudo-ttys, and shared memory.

While the format is admittedly not very user-friendly, looking at **`/proc/mounts`** is the best way to be 100% sure of seeing what is mounted on your Red Hat Enterprise Linux system, as the kernel is providing this information. Other methods can, under rare circumstances, be inaccurate.

However, most of the time you will likely use a command with more easily-read (and useful) output. The next section describes that command.

5.9.3.2.3. Issuing the `df` Command

While using **`/etc/mtab`** or **`/proc/mounts`** lets you know what file systems are currently mounted, it does little beyond that. Most of the time you are more interested in one particular aspect of the file systems that are currently mounted -- the amount of free space on them.

For this, we can use the **`df`** command. Here is some sample output from **`df`**:

```
Filesystem 1k-blocks Used Available Use% Mounted on /dev/sda3 8428196 4280980 3719084
54% / /dev/sda1 124427 18815 99188 16% /boot /dev/sda4 8428196 4094232 3905832 52% /home
none 644600 0 644600 0% /dev/shm
```

Several differences from **`/etc/mtab`** and **`/proc/mount`** are immediately obvious:

- An easy-to-read heading is displayed
- With the exception of the shared memory file system, only disk-based file systems are shown
- Total size, used space, free space, and percentage in use figures are displayed

That last point is probably the most important because every system administrator eventually has to deal with a system that has run out of free disk space. With **`df`** it is very easy to see where the problem lies.

5.9.4. Network-Accessible Storage Under Red Hat Enterprise Linux

There are two major technologies used for implementing network-accessible storage under Red Hat Enterprise Linux:

- NFS

- SMB

The following sections describe these technologies.

5.9.4.1. NFS

As the name implies, the Network File System (more commonly known as NFS) is a file system that may be accessed via a network connection. With other file systems, the storage device must be directly attached to the local system. However, with NFS this is not a requirement, making possible a variety of different configurations, from centralized file system servers to entirely diskless computer systems.

However, unlike the other file systems, NFS does not dictate a specific on-disk format. Instead, it relies on the server operating system's native file system support to control the actual I/O to local disk drive(s). NFS then makes the file system available to any operating system running a compatible NFS client.

While primarily a Linux and UNIX technology, it is worth noting that NFS client implementations exist for other operating systems, making NFS a viable technique to share files with a variety of different platforms.

The file systems an NFS server makes available to clients is controlled by the configuration file **/etc/exports**. For more information, see the **exports(5)** man page and the *System Administrators Guide*.

5.9.4.2. SMB

SMB stands for *Server Message Block* and is the name for the communications protocol used by various operating systems produced by Microsoft over the years. SMB makes it possible to share storage across a network. Present-day implementations often use TCP/IP as the underlying transports; previously NetBEUI was the transport.

Red Hat Enterprise Linux supports SMB via the Samba server program. The *System Administrators Guide* includes information on configuring Samba.

5.9.5. Mounting File Systems Automatically with /etc/fstab

When a Red Hat Enterprise Linux system is newly-installed, all the disk partitions defined and/or created during the installation are configured to be automatically mounted whenever the system boots. However, what happens when additional disk drives are added to a system after the installation is done? The answer is "nothing" because the system was not configured to mount them automatically. However, this is easily changed.

The answer lies in the **/etc/fstab** file. This file is used to control what file systems are mounted when the system boots, as well as to supply default values for other file systems that may be mounted manually from time to time. Here is a sample **/etc/fstab** file:

```
LABEL=/ / ext3 defaults 1 1 /dev/sda1 /boot ext3 defaults 1 2 /dev/cdrom /mnt/cdrom iso9660
noauto,owner,kudzu,ro 0 0 /dev/homedisk /home ext3 defaults 1 2 /dev/sda2 swap swap defaults
0 0
```

Each line represents one file system and contains the following fields:

- File system specifier -- For disk-based file systems, either a device file name (**/dev/sda1**), a file system label specification (**LABEL=/**), or a **devlabel**-managed symbolic link (**/dev/homedisk**)

- Mount point -- Except for swap partitions, this field specifies the mount point to be used when the file system is mounted (**`/boot`**)

- File system type -- The type of file system present on the specified device (note that **`auto`** may be specified to select automatic detection of the file system to be mounted, which is handy for removable media units such as diskette drives)

- Mount options -- A comma-separated list of options that can be used to control **`mount`**'s behavior (**`noauto,owner,kudzu`**)

- Dump frequency -- If the **`dump`** backup utility is used, the number in this field controls **`dump`**'s handling of the specified file system

- File system check order -- Controls the order in which the file system checker **`fsck`** checks the integrity of the file systems

5.9.6. Adding/Removing Storage

While most of the steps required to add or remove storage depend more on the system hardware than the system software, there are aspects of the procedure that are specific to your operating environment. This section explores the steps necessary to add and remove storage that are specific to Red Hat Enterprise Linux.

5.9.6.1. Adding Storage

The process of adding storage to a Red Hat Enterprise Linux system is relatively straightforward. Here are the steps that are specific to Red Hat Enterprise Linux:

- Partitioning

- Formatting the partition(s)

- Updating **`/etc/fstab`**

The following sections explore each step in more detail.

5.9.6.1.1. Partitioning

Once the disk drive has been installed, it is time to create one or more partitions to make the space available to Red Hat Enterprise Linux.

There is more than one way of doing this:

- Using the command-line **`fdisk`** utility program

- Using **`parted`**, another command-line utility program

Although the tools may be different, the basic steps are the same. In the following example, the commands necessary to perform these steps using **`fdisk`** are included:

1. Select the new disk drive (the drive's name can be identified by following the device naming conventions outlined in *Section 5.9.1, “Device Naming Conventions”*). Using **`fdisk`**, this is done by including the device name when you start **`fdisk`**:

   ```
   fdisk /dev/hda
   ```

2. View the disk drive's partition table, to ensure that the disk drive to be partitioned is, in fact, the correct one. In our example, **`fdisk`** displays the partition table by using the **`p`** command:

   ```
   Command (m for help): p Disk /dev/hda: 255 heads, 63 sectors, 1244 cylinders Units =
   cylinders of 16065 * 512 bytes Device Boot Start End Blocks Id System /dev/hda1 * 1 17
   136521 83 Linux /dev/hda2 18 83 530145 82 Linux swap /dev/hda3 84 475 3148740 83 Linux /
   dev/hda4 476 1244 6176992+ 83 Linux
   ```

3. Delete any unwanted partitions that may already be present on the new disk drive. This is done using the **`d`** command in **`fdisk`**:

   ```
   Command (m for help): d Partition number (1-4): 1
   ```

   The process would be repeated for all unneeded partitions present on the disk drive.

4. Create the new partition(s), being sure to specify the desired size and file system type. Using **`fdisk`**, this is a two-step process -- first, creating the partition (using the **`n`** command):

   ```
   Command (m for help): n Command action e extended p primary partition (1-4) p
   Partition number (1-4): 1 First cylinder (1-767): 1 Last cylinder or +size or +sizeM or
   +sizeK: +512M
   ```

   Second, by setting the file system type (using the **`t`** command):

   ```
   Command (m for help): t Partition number (1-4): 1 Hex code (type L to list codes): 82
   ```

   Partition type 82 represents a Linux swap partition.

5. Save your changes and exit the partitioning program. This is done in **`fdisk`** by using the **`w`** command:

   ```
   Command (m for help): w
   ```

Warning

When partitioning a new disk drive, it is *vital* that you are sure the disk drive you are about to partition is the correct one. Otherwise, you may inadvertently partition a disk drive that is already in use, resulting in lost data.

Also make sure you have decided on the best partition size. Always give this matter serious thought, because changing it later is much more difficult than taking a bit of time now to think things through.

5.9.6.1.2. Formatting the Partition(s)

Formatting partitions under Red Hat Enterprise Linux is done using the **`mkfs`** utility program. However, **`mkfs`** does not actually do the work of writing the file-system-specific information onto a disk drive; instead it passes control to one of several other programs that actually create the file system.

This is the time to look at the **mkfs.*<fstype>*** man page for the file system you have selected. For example, look at the **mkfs.ext3** man page to see the options available to you when creating a new ext3 file system. In general, the **mkfs.*<fstype>*** programs provide reasonable defaults for most configurations; however here are some of the options that system administrators most commonly change:

- Setting a volume label for later use in **/etc/fstab**
- On very large hard disks, setting a lower percentage of space reserved for the super-user
- Setting a non-standard block size and/or bytes per inode for configurations that must support either very large or very small files
- Checking for bad blocks before formatting

Once file systems have been created on all the appropriate partitions, the disk drive is properly configured for use.

Next, it is always best to double-check your work by manually mounting the partition(s) and making sure everything is in order. Once everything checks out, it is time to configure your Red Hat Enterprise Linux system to automatically mount the new file system(s) whenever it boots.

5.9.6.1.3. Updating /etc/fstab

As outlined in *Section 5.9.5, “Mounting File Systems Automatically with /etc/fstab”*, you must add the necessary line(s) to **/etc/fstab** to ensure that the new file system(s) are mounted whenever the system reboots. Once you have updated **/etc/fstab**, test your work by issuing an "incomplete" **mount**, specifying only the device or mount point. Something similar to one of the following commands is sufficient:

```
mount /home mount /dev/hda3
```

(Replacing **/home** or **/dev/hda3** with the mount point or device for your specific situation.)

If the appropriate **/etc/fstab** entry is correct, **mount** obtains the missing information from it and completes the mount operation.

At this point you can be relatively confident that **/etc/fstab** is configured properly to automatically mount the new storage every time the system boots (although if you can afford a quick reboot, it would not hurt to do so -- just to be sure).

5.9.6.2. Removing Storage

The process of removing storage from a Red Hat Enterprise Linux system is relatively straightforward. Here are the steps that are specific to Red Hat Enterprise Linux:

- Remove the disk drive's partitions from **/etc/fstab**
- Unmount the disk drive's active partitions
- Erase the contents of the disk drive

The following sections cover these topics in more detail.

5.9.6.2.1. Remove the Disk Drive's Partitions From `/etc/fstab`

Using the text editor of your choice, remove the line(s) corresponding to the disk drive's partition(s) from the **`/etc/fstab`** file. You can identify the proper lines by one of the following methods:

- Matching the partition's mount point against the directories in the second column of **`/etc/fstab`**
- Matching the device's file name against the file names in the first column of **`/etc/fstab`**

> **Note**
>
> Be sure to look for any lines in **`/etc/fstab`** that identify swap partitions on the disk drive to be removed; they can be easily overlooked.

5.9.6.2.2. Terminating Access With `umount`

Next, all access to the disk drive must be terminated. For partitions with active file systems on them, this is done using the **`umount`** command. If a swap partition exists on the disk drive, it must be either be deactivated with the **`swapoff`** command, or the system should be rebooted.

Unmounting partitions with the **`umount`** command requires you to specify either the device file name, or the partition's mount point:

```
umount /dev/hda2 umount /home
```

A partition can only be unmounted if it is not currently in use. If the partition cannot be unmounted while at the normal runlevel, boot into rescue mode and remove the partition's **`/etc/fstab`** entry.

When using **`swapoff`** to disable swapping to a partition, you must specify the device file name representing the swap partition:

```
swapoff /dev/hda4
```

If swapping to a swap partition cannot be disabled using **`swapoff`**, boot into rescue mode and remove the partition's **`/etc/fstab`** entry.

5.9.6.2.3. Erase the Contents of the Disk Drive

Erasing the contents of a disk drive under Red Hat Enterprise Linux is a straightforward procedure.

After unmounting all of the disk drive's partitions, issue the following command (while logged in as root):

```
badblocks -ws <device-name>
```

Where ***`<device-name>`*** represents the file name of the disk drive you wish to erase, excluding the partition number. For example, **`/dev/hdb`** for the second ATA hard drive.

The following output is displayed while **`badblocks`** runs:

```
Writing pattern 0xaaaaaaaa: done Reading and comparing: done Writing pattern 0x55555555:
done Reading and comparing: done Writing pattern 0xffffffff: done Reading and comparing:
done Writing pattern 0x00000000: done Reading and comparing: done
```

Keep in mind that **badblocks** is actually writing four different data patterns to every block on the disk drive. For large disk drives, this process can take a long time -- quite often several hours.

Important

Many companies (and government agencies) have specific methods of erasing data from disk drives and other data storage media. You should *always* be sure you understand and abide by these requirements; in many cases there are legal ramifications if you fail to do so. The example above should in no way be considered the ultimate method of wiping a disk drive.

However, it is much more effective than using the **rm** command. That is because when you delete a file using **rm** it only marks the file as deleted -- it does *not* erase the contents of the file.

5.9.7. Implementing Disk Quotas

Red Hat Enterprise Linux is capable of keeping track of disk space usage on a per-user and per-group basis through the use of disk quotas. The following section provides an overview of the features present in disk quotas under Red Hat Enterprise Linux.

5.9.7.1. Some Background on Disk Quotas

Disk quotas under Red Hat Enterprise Linux have the following features:

- Per-file-system implementation
- Per-user space accounting
- Per-group space accounting
- Tracks disk block usage
- Tracks disk inode usage
- Hard limits
- Soft limits
- Grace periods

The following sections describe each feature in more detail.

5.9.7.1.1. Per-File-System Implementation

Disk quotas under Red Hat Enterprise Linux can be used on a per-file-system basis. In other words, disk quotas can be enabled or disabled for each file system individually.

This provides a great deal of flexibility to the system administrator. For example, if the **/home/** directory was on its own file system, disk quotas could be enabled there, enforcing equitable disk usage by all users. However the root file system could be left without disk quotas, eliminating the complexity of maintaining disk quotas on a file system where only the operating system itself resides.

5.9.7.1.2. Per-User Space Accounting

Disk quotas can perform space accounting on a per-user basis. This means that each user's space usage is tracked individually. It also means that any limitations on usage (which are discussed in later sections) are also done on a per-user basis.

Having the flexibility of tracking and enforcing disk usage for each user individually makes it possible for a system administrator to assign different limits to individual users, according to their responsibilities and storage needs.

5.9.7.1.3. Per-Group Space Accounting

Disk quotas can also perform disk usage tracking on a per-group basis. This is ideal for those organizations that use groups as a means of combining different users into a single project-wide resource.

By setting up group-wide disk quotas, the system administrator can more closely manage storage utilization by giving individual users only the disk quota they require for their personal use, while setting larger disk quotas that would be more appropriate for multi-user projects. This can be a great advantage to those organizations that use a "chargeback" mechanism to assign data center costs to those departments and teams that use data center resources.

5.9.7.1.4. Tracks Disk Block Usage

Disk quotas track disk block usage. Because all the data stored on a file system is stored in blocks, disk quotas are able to directly correlate the files created and deleted on a file system with the amount of storage those files take up.

5.9.7.1.5. Tracks Disk Inode Usage

In addition to tracking disk block usage, disk quotas also can track inode usage. Under Red Hat Enterprise Linux, inodes are used to store various parts of the file system, but most importantly, inodes hold information for each file. Therefore, by tracking (and controlling) inode usage, it is possible to control the creation of new files.

5.9.7.1.6. Hard Limits

A hard limit is the absolute maximum number of disk blocks (or inodes) that can be temporarily used by a user (or group). Any attempt to use a single block or inode above the hard limit fails.

5.9.7.1.7. Soft Limits

A soft limit is the maximum number of disk blocks (or inodes) that can be permanently used by a user (or group).

The soft limit is set below the hard limit. This allows users to temporarily exceed their soft limit, permitting them to finish whatever they were doing, and giving them some time in which to go through their files and trim back their usage to below their soft limit.

5.9.7.1.8. Grace Periods

As stated earlier, any disk usage above the soft limit is temporary. It is the grace period that determines the length of time that a user (or group) can extend their usage beyond their soft limit and toward their hard limit.

If a user continues to use more than the soft limit and the grace period expires, no additional disk usage will be permitted until the user (or group) has reduced their usage to a point below the soft limit.

The grace period can be expressed in seconds, minutes, hours, days, weeks, or months, giving the system administrator a great deal of freedom in determining how much time to give users to get their disk usages below their soft limits.

5.9.7.2. Enabling Disk Quotas

Note

The following sections provide a brief overview of the steps necessary to enable disk quotas under Red Hat Enterprise Linux. For a more in-depth treatment of this subject, see the chapter on disk quotas in the *System Administrators Guide*.

To use disk quotas, you must first enable them. This process involves several steps:

1. Modifying **`/etc/fstab`**

2. Remounting the file system(s)

3. Running **`quotacheck`**

4. Assigning quotas

The **`/etc/fstab`** file controls the mounting of file systems under Red Hat Enterprise Linux. Because disk quotas are implemented on a per-file-system basis, there are two options -- **`usrquota`** and **`grpquota`** -- that must be added to that file to enable disk quotas.

The **`usrquota`** option enables user-based disk quotas, while the **`grpquota`** option enables group-based quotas. One or both of these options may be enabled by placing them in the options field for the desired file system.

The affected file system(s) then must be unmounted and remounted for the disk quota-related options to take affect.

Next, the **`quotacheck`** command is used to create the disk quota files and to collect the current usage information from already existing files. The disk quota files (named **`aquota.user`** and **`aquota.group`** for user- and group-based quotas) contain the necessary quota-related information and reside in the file system's root directory.

To assign disk quotas, the **`edquota`** command is used.

This utility program uses a text editor to display the quota information for the user or group specified as part of the **`edquota`** command. Here is an example:

```
Disk quotas for user matt (uid 500): Filesystem blocks soft hard inodes soft hard /dev/md3
6618000 0 0 17397 0 0
```

This shows that user matt is currently using over 6GB of disk space, and over 17,000 inodes. No quota (soft or hard) has yet been set for either disk blocks or inodes, meaning that there is no limit to the disk space and inodes that this user can currently use..

Using the text editor displaying the disk quota information, the system administrator can then modify the soft and hard limits as desired:

```
Disk quotas for user matt (uid 500): Filesystem blocks soft hard inodes soft hard /dev/md3
6618000 6900000 7000000 17397 0 0
```

In this example, user matt has been given a soft limit of 6.9GB and a hard limit of 7GB. No soft or hard limit on inodes has been set for this user.

Note

The **`edquota`** program can also be used to set the per-file-system grace period by using the **`-t`** option.

5.9.7.3. Managing Disk Quotas

There is little actual management required to support disk quotas under Red Hat Enterprise Linux. Essentially, all that is required is:

- Generating disk usage reports at regular intervals (and following up with users that seem to be having trouble effectively managing their allocated disk space)

- Making sure that the disk quotas remain accurate

Creating a disk usage report entails running the **`repquota`** utility program. Using the command **`repquota /home`** produces this output:

```
*** Report for user quotas on device /dev/md3 Block grace time: 7days; Inode grace
time: 7days Block limits File limits User used soft hard grace used soft hard grace
---------------------------------------------------------------------- root -- 32836 0 0 4 0
0 matt -- 6618000 6900000 7000000 17397 0 0
```

More information about **`repquota`** can be found in the *System Administrators Guide*, in the chapter on disk quotas.

Whenever a file system is not unmounted cleanly (due to a system crash, for example), it is necessary to run **`quotacheck`**. However, many system administrators recommend running **`quotacheck`** on a regular basis, even if the system has not crashed.

The process is similar to the initial use of **`quotacheck`** when enabling disk quotas.

Here is an example **`quotacheck`** command:

```
quotacheck -avug
```

The easiest way to run **`quotacheck`** on a regular basis is to use **`cron`**. Most system administrators run **`quotacheck`** once a week, though there may be valid reasons to pick a longer or shorter interval, depending on your specific conditions.

5.9.8. Creating RAID Arrays

In addition to supporting hardware RAID solutions, Red Hat Enterprise Linux supports software RAID. There are two ways that software RAID arrays can be created:

- While installing Red Hat Enterprise Linux
- After Red Hat Enterprise Linux has been installed

The following sections review these two methods.

5.9.8.1. While Installing Red Hat Enterprise Linux

During the normal Red Hat Enterprise Linux installation process, RAID arrays can be created. This is done during the disk partitioning phase of the installation.

To begin, you must manually partition your disk drives using **Disk Druid**. You must first create a new partition of the type "software RAID." Next, select the disk drives that you want to be part of the RAID array in the **Allowable Drives** field. Continue by selecting the desired size and whether you want the partition to be a primary partition.

Once you have created all the partitions required for the RAID array(s) that you want to create, you must then use the **RAID** button to actually create the arrays. You are then presented with a dialog box where you can select the array's mount point, file system type, RAID device name, RAID level, and the "software RAID" partitions on which this array is to be based.

Once the desired arrays have been created, the installation process continues as usual.

Note

For more information on creating software RAID arrays during the Red Hat Enterprise Linux installation process, refer to the *System Administrators Guide*.

5.9.8.2. After Red Hat Enterprise Linux Has Been Installed

Creating a RAID array after Red Hat Enterprise Linux has been installed is a bit more complex. As with the addition of any type of disk storage, the necessary hardware must first be installed and properly configured.

Partitioning is a bit different for RAID than it is for single disk drives. Instead of selecting a partition type of "Linux" (type 83) or "Linux swap" (type 82), all partitions that are to be part of a RAID array must be set to "Linux raid auto" (type fd).

Next, it is necessary to actually create the RAID array. This is done with the **`mdadm`** program (refer to **`man mdadm`** for more information).

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/hd[bc]1
mdadm --detail --scan > /dev/mdadm.conf
```

The RAID array **`/dev/md0`** is now ready to be formatted and mounted. The process at this point is no different than for formatting and mounting a single disk drive.

5.9.9. Day to Day Management of RAID Arrays

There is little that needs to be done to keep a RAID array operating. As long as no hardware problems crop up, the array should function just as if it were a single physical disk drive. However, just as a system administrator should periodically check the status of all disk drives on the system, the RAID arrays' status should be checked as well.

5.9.9.1. Checking Array Status With `/proc/mdstat`

The file **`/proc/mdstat`** is the easiest way to check on the status of all RAID arrays on a particular system. Here is a sample **`mdstat`** (view with the command **`cat /proc/mdstat`**):

```
Personalities : [raid1] read_ahead 1024 sectors md1 : active raid1 hda3[0] hdc3[1] 522048
blocks [2/2] [UU] md0 : active raid1 hda2[0] hdc2[1] 4192896 blocks [2/2] [UU] md2 : active
raid1 hda1[0] hdc1[1] 128384 blocks [2/2] [UU] unused devices: <none>
```

On this system, there are three RAID arrays (all RAID 1). Each RAID array has its own section in **`/proc/mdstat`** and contains the following information:

- The RAID array device name (not including the **`/dev/`** part)
- The status of the RAID array
- The RAID array's RAID level
- The physical partitions that currently make up the array (followed by the partition's array unit number)
- The size of the array
- The number of configured devices versus the number of operative devices in the array
- The status of each configured device in the array (**`U`** meaning the device is OK, and _ indicating that the device has failed)

5.9.9.2. Rebuilding a RAID array

Should **`/proc/mdstat`** show that a problem exists with one of the RAID arrays, you can rebuild it by performing the following steps:

1. Remove the disk from the raid array.

   `mdadm --manage /dev/md0 -r /dev/sdc3`

2. Remove the disk from the system.

3. Using **`fdisk`**, replace the removed disk and re-format the replacement disk.

4. Add the new disk back to the RAID array.

   `mdadm --manage /dev/md0 -a /dev/sdc3`

5. To restore the disk, perform a "software fail" the previous spare slice:

   `mdadm --manage --set-faulty /dev/md0 /dev/sdc3`

6. The system will now attempt to rebuild the array on the replaced disk. Use the following command to monitor status:

   `watch -n 1 cat /proc/mdstat`

7. When the array is finished rebuilding, remove and then re-add the software-failed disk back to the array.

   `mdadm --manage /dev/md0 -r /dev/sdc3`

```
mdadm --manage /dev/md0 -a /dev/sdc3
```

8. Check the array.

```
mdadm --detail /dev/md0
```

5.9.10. Logical Volume Management

Red Hat Enterprise Linux includes support for LVM. LVM may be configured while Red Hat Enterprise Linux is installed, or it may be configured after the installation is complete. LVM under Red Hat Enterprise Linux supports physical storage grouping, logical volume resizing, and the migration of data off a specific physical volume.

For more information on LVM, refer to the *System Administrators Guide*.

5.10. Additional Resources

This section includes various resources that can be used to learn more about storage technologies and the Red Hat Enterprise Linux-specific subject matter discussed in this chapter.

5.10.1. Installed Documentation

The following resources are installed in the course of a typical Red Hat Enterprise Linux installation, and can help you learn more about the subject matter discussed in this chapter.

- **`exports(5)`** man page -- Learn about the NFS configuration file format.
- **`fstab(5)`** man page -- Learn about the file system information configuration file format.
- **`swapoff(8)`** man page -- Learn how to disable swap partitions.
- **`df(1)`** man page -- Learn how to display disk space usage on mounted file systems.
- **`fdisk(8)`** man page -- Learn about this partition table maintenance utility program.
- **`mkfs(8)`**, **`mke2fs(8)`** man pages -- Learn about these file system creation utility programs.
- **`badblocks(8)`** man page -- Learn how to test a device for bad blocks.
- **`quotacheck(8)`** man page -- Learn how to verify block and inode usage for users and groups and optionally creates disk quota files.
- **`edquota(8)`** man page -- Learn about this disk quota maintenance utility program.
- **`repquota(8)`** man page -- Learn about this disk quota reporting utility program.
- **`raidtab(5)`** man page -- Learn about the software RAID configuration file format.
- **`mdadm(8)`** man page -- Learn about this software RAID array management utility program.
- **`lvm(8)`** man page -- Learn about Logical Volume Management.
- **`devlabel(8)`** man page -- Learn about persistent storage device access.

5.10.2. Useful Websites

- *http://www.pcguide.com/* -- A good site for all kinds of information on various storage technologies.

- *http://www.tldp.org/* -- The Linux Documentation Project has HOWTOs and mini-HOWTOs that provide good overviews of storage technologies as they relate to Linux.

5.10.3. Related Books

The following books discuss various issues related to storage and are good resources for Red Hat Enterprise Linux system administrators.

- The *Installation Guide*; Red Hat, Inc -- Contains instructions on partitioning hard drives during the Red Hat Enterprise Linux installation process as well as an overview of disk partitions.
- The *Reference Guide*; Red Hat, Inc -- Contains detailed information on the directory structure used in Red Hat Enterprise Linux and an overview of NFS.
- The *System Administrators Guide*; Red Hat, Inc -- Includes chapters on file systems, RAID, LVM, **`devlabel`**, partitioning, disk quotas, NFS, and Samba.
- *Linux System Administration: A User's Guide* by Marcel Gagne; Addison Wesley Professional -- Contains information on user and group permissions, file systems and disk quota, NFS and Samba.
- *Linux Performance Tuning and Capacity Planning* by Jason R. Fink and Matthew D. Sherer; Sams -- Contains information on disk, RAID, and NFS performance.
- *Linux Administration Handbook* by Evi Nemeth, Garth Snyder, and Trent R. Hein; Prentice Hall -- Contains information on file systems, handling disk drives, NFS, and Samba.

Managing User Accounts and Resource Access

Managing *user accounts* and *groups* is an essential part of system administration within an organization. But to do this effectively, a good system administrator must first understand what user accounts and groups are and how they work.

The primary reason for user accounts is to verify the identity of each individual using a computer system. A secondary (but still important) reason for user accounts is to permit the per-individual tailoring of resources and access privileges.

Resources can include files, directories, and devices. Controlling access to these resources is a large part of a system administrator's daily routine; often the access to a resource is controlled by groups. Groups are logical constructs that can be used to cluster user accounts together for a common purpose. For example, if an organization has multiple system administrators, they can all be placed in one system administrator group. The group can then be given permission to access key system resources. In this way, groups can be a powerful tool for managing resources and access.

The following sections discuss user accounts and groups in more detail.

6.1. Managing User Accounts

As stated earlier, user accounts are the method by which an individual is identified and authenticated to the system. User accounts have several different components to them. First, there is the username. The password is next, followed by the access control information.

The following sections explore each of these components in more detail.

6.1.1. The Username

From the system's standpoint, the username is the answer to the question, "who are you?" As such, usernames have one major requirement: they must be unique. In other words, each user must have a username that is different from all other usernames on that system.

Because of this requirement, it is vital to determine (in advance) how usernames are to be created. Otherwise, you may find yourself in the position of being forced to react each time a new user requests an account.

What you need is a naming convention for your user accounts.

6.1.1.1. Naming Conventions

By creating a naming convention for usernames, you can save yourself a great deal of trouble. Instead of making up names as you go along (and finding it harder and harder to come up with a reasonable name), you do some work up-front and devise a convention to be used for all subsequent user accounts. Your naming convention can be very simple, or the description alone could take several pages to document.

The exact nature of your naming convention should take several factors into account:

- The size of your organization
- The structure of your organization
- The nature of your organization

The size of your organization matters, as it dictates how many users your naming convention must support. For example, a very small organization might be able to have everyone use their first name. For a much larger organization this naming convention would not work.

An organization's structure can also have a bearing on the most appropriate naming convention. For organizations with a strictly-defined structure it might be appropriate to include elements of that structure in the naming convention. For example, you could include your organization's departmental codes as part of each username.

The overall nature of your organization may also mean that some naming conventions are more appropriate than others. An organization that deals with highly-classified data might choose a naming convention that does away with any personally-identifiable ties between the individual and their name. In such an organization, Maggie McOmie's username might be LUH3417.

Here are some naming conventions that other organizations have used:

- First name (john, paul, george, etc.)
- Last name (smith, jones, brown, etc.)
- First initial, followed by last name (jsmith, pjones, gbrown, etc.)
- Last name, followed by department code (smith029, jones454, brown191, etc.)

Note

Be aware that if your naming convention includes appending different data together to form a username, the potential exists that the result might be offensive or humorous. Therefore, even if you have automated username creation, it is wise to have some sort of review process in place.

One thing in common with the naming conventions described here is that it is possible that eventually there will be two individuals that, according to the naming convention, should be given the same username. This is known as a *collision*. Because each username must be unique, it is necessary to address the issue of collisions. The following section does this.

6.1.1.1.1. Dealing with Collisions

Collisions are a fact of life — no matter how you try, you will eventually find yourself dealing with a collision. You must plan for collisions in your naming convention. There are several ways this can be done:

- Adding sequence numbers to the colliding username (smith, smith1, smith2, etc.)
- Adding user-specific data to the colliding username (smith, esmith, eksmith, etc.)
- Adding organizational information to the colliding username (smith, smith029, smith454, etc.)

Having some method of resolving collisions is a necessary part of any naming convention. However, it does make it more difficult for someone outside the organization to accurately determine an individual's username. Therefore, the downside of most naming conventions is that the occasional misdirected email becomes more likely.

6.1.1.2. Dealing with Name Changes

If your organization uses a naming convention that is based on each user's name, it is a fact of life that you will eventually have to deal with name changes. Even if a person's actual name does not change, a change in username may from time to time be requested. The reasons can range from the user not being satisfied with the username to the user being a senior official in your organization and willing to use their influence to obtain a "more appropriate" username.

No matter what the reason, there are several issues to keep in mind when changing a username:

- Make the change to *all* affected systems
- Keep any underlying user identification constant
- Change the ownership of all files and other user-specific resources (if necessary)
- Handle email-related issues

First and foremost, it is important to make sure that the new username is propagated to all systems where the original username was in use. Otherwise, any operating system function that relies on the username may work on some systems and not on others. Certain operating systems use access control techniques based on usernames; such systems are particularly vulnerable to problems stemming from a changed username.

Many operating systems use some sort of user identification number for most user-specific processing. To minimize the problems stemming from a username change, try to keep this identification number constant between the new and the old username. Failure to do so often results in a scenario where the user can no longer access files and other resources that they had previously owned under their original username.

If the user identification number must be changed, it is necessary to change the ownership for all files and user-specific resources to reflect the new user identification. This can be an error-prone process, as it seems that there is always something in some forgotten corner of a system that ends up being overlooked.

Issues related to email are probably the one area where a username change is the most difficult. The reason for this is that unless steps are taken to counteract it, email addressed to the old username will not be delivered to the new username.

Unfortunately, the issues surrounding the impact of username changes on email are multi-dimensional. At its most basic, a username change means that people no longer know the correct username for the person. At first glance, this might not seem to be such a problem — notify everyone in your organization of the change. But what about anyone outside of your organization that has sent this person email? How should they be notified? And what about mailing lists (both internal and external)? How can they be updated?

There is no easy answer to these questions. The best answer may be one of creating an email alias such that all email sent to the old username is automatically forwarded to the new username. The user can then be urged to alert anyone who sends them email that their username has changed. As time goes on, fewer and fewer email messages will be delivered using the alias; eventually the alias can be removed.

While the use of aliases, at some level, perpetuates an incorrect assumption (that the user now known as esmith is still known as ejones), it is the only way to guarantee that email reaches the proper person.

Important

If you use email aliases, be sure you take whatever steps are necessary to protect the old username from potential reuse. If you do not do this, and a new user receives the old username, email delivery (for either the original user or the new user) may be disrupted. The exact nature of the disruption depends on how email delivery is implemented on your operating system, but the two most likely symptoms are:

- The new user never receives any email — it all goes to the original user.
- The original user suddenly stops receiving any email — it all goes to the new user.

6.1.2. Passwords

If the username provides an answer to the question, "who are you?", the password is the response to the demand that inevitably follows:

"Prove it!"

In more formal terms, a password provides a means of proving the authenticity of a person's claim to be the user indicated by the username. The effectiveness of a password-based authentication scheme relies heavily on several aspects of the password:

- The secrecy of the password
- The resistance of the password to guessing
- The resistance of the password to a brute-force attack

Passwords that adequately address these issues are said to be *strong*, while those that fail to address one or more of these issues is said to be *weak*. Creating strong passwords is important for the security of the organization, as strong passwords are less likely to be discovered or guessed. There are two options available to enforce the use of strong passwords:

- The system administrator can create passwords for all users.
- The system administrator can let the users create their own passwords, while verifying that the passwords are acceptably strong.

Creating passwords for all users ensures that the passwords are strong, but it becomes a daunting task as the organization grows. It also increases the risk of users writing their passwords down.

For these reasons, most system administrators prefer to have their users create their own passwords. However, a good system administrator takes steps to verify that the passwords are strong.

For guidelines on creating strong passwords, see the chapter titled *Workstation Security* in the *Red Hat Enterprise Linux Security Guide*.

The need for passwords to be kept secret should be an ingrained part of every system administrator's mindset. However, this point is often lost on many users. In fact, many users do not even understand the difference between usernames and passwords. Given this unfortunate fact of life, it is vital that some amount of user education be undertaken, so that your users understand that their password should be kept as secret as their paycheck.

Passwords should be as difficult as possible to guess. A strong password is one that an attacker would not be able to guess, even if the attacker knew the user well.

A brute-force attack on a password entails methodically trying (usually via a program known as a *password-cracker*) every possible combination of characters in the hopes that the correct password will eventually be found. A strong password should be constructed in such a way as to make the number of potential passwords that must be tested very large, forcing the attacker to take a long time searching for the password.

Strong and weak passwords are explored in more detail in the following sections.

6.1.2.1. Weak Passwords

As stated earlier, a weak password fails one of these three tests:

- It is secret
- It is resistant to being guessed
- It is resistant to a brute-force attack

The following sections show how passwords can be weak.

6.1.2.1.1. Short Passwords

A password that is short is weak because it much more susceptible to a brute-force attack. To illustrate this, consider the following table, where the number of potential passwords that would have to be tested in a brute-force attack is shown. (The passwords are assumed to consist only of lower-case letters.)

Table 6.1. Password Length Versus the Number of Potential Passwords

Password Length	Potential Passwords
1	26
2	676
3	17,576
4	456,976
5	11,881,376
6	308,915,776

As you can see, the number of possible passwords increases dramatically as the length increases.

Note

Even though this table ends at six characters, this should not be construed as recommending that six-character passwords are sufficiently long for good security. In general, the longer the password, the better.

6.1.2.1.2. Limited Character Set

The number of different characters that can comprise a password has a large impact on the ability of an attacker to conduct a brute-force attack. For example, instead of the 26 different characters that can be used in a lower-case-only password, what if we also used digits? That would mean each character in a password could be one of 36 characters instead of just one of 26. In the case of a six-character password, this increases the number of possible passwords from 308,915,776 to 2,176,782,336.

There is still more that can be done. If we also include mixed-case alphanumeric passwords (for those operating systems that support it), the number of possible six-character passwords increases to 56,800,235,584. Adding other characters (such as punctuation marks) further increases the number of possible passwords, making a brute-force attack that much more difficult.

However, one point to keep in mind is that not every attack against a password is a brute-force attack. The following sections describe other attributes that can make a weak password.

6.1.2.1.3. Recognizable Words

Many attacks against passwords are based on the fact that people are most comfortable with passwords they can remember. And for most people, passwords that are memorable are passwords that contain words. Therefore, most password attacks are dictionary-based. In other words, the attacker uses dictionaries of words in an attempt to find the word or words that comprise a password.

> **Note**
>
> Many dictionary-based password attack programs use dictionaries from multiple languages. Therefore, you should not feel that you have a strong password just because you have used non-English words in your password.

6.1.2.1.4. Personal Information

Passwords that contain personal information (the name or birth date of a loved one, a pet, or a personal identification number) may or may not be picked up by a dictionary-based password attack. However, if the attacker knows you personally (or is sufficiently motivated to research your personal life), they might be able to guess your password with little or no difficulty.

In addition to dictionaries, many password-crackers also include common names, dates, and other such information in their search for passwords. Therefore, even if the attacker does not know that your dog is named Gracie, they could still find out that your password is "mydogisgracie", with a good password-cracker.

6.1.2.1.5. Simple Word Tricks

Using any of the previously discussed information as the basis for a password, but reversing the character order does not turn a weak password into a strong password. Most password-crackers perform such tricks on possible passwords. This includes substituting certain numbers for letters in common words. Here are some examples:

- drowssaPdaB1
- R3allyP00r

6.1.2.1.6. The Same Password for Multiple Systems

Even if you have a password that is strong, it is a bad idea to use the exact same password on more than one system. Obviously little can be done if the systems are configured to use a central authentication server of some kind, but in every other instance, different passwords should be used for each system.

6.1.2.1.7. Passwords on Paper

Another way to turn a strong password into a weak one is to write it down. By putting a password on paper, you no longer have a secrecy problem, you have a physical security problem — now you must keep a piece of paper secure. Therefore, writing down a password is never a good idea.

However, some organizations have a legitimate need for written passwords. For example, some organizations have written passwords as part of a procedure to recover from the loss of key personnel (such as system administrators). In these instances, the paper containing the passwords is stored in a physically-secure location that requires multiple people to cooperate in order to get access to the paper. Vaults with multiple locks and bank safe deposit boxes are often used.

Any organization that explores this method of storing passwords for emergency purposes should be aware that the existence of written passwords adds an element of risk to their systems' security, no matter how securely the written passwords may be stored. This is particularly true if it is generally known that the passwords are written down (and where they are stored).

Unfortunately, written passwords are often not part of a recovery plan and are not stored in a vault, but are passwords for ordinary users, and are stored in the following places:

- In a desk drawer (locked or unlocked)
- Below a keyboard
- In a wallet
- Taped to the side of a monitor

None of these locations are proper places for a written password.

6.1.2.2. Strong Passwords

We have seen what weak passwords are like; the following sections describe features that all strong passwords possess.

6.1.2.2.1. Longer Passwords

The longer a password is, the less likely it is that a brute-force attack may succeed. Therefore, if your operating system supports it, set relatively large minimum password lengths for your users.

6.1.2.2.2. Expanded Character Set

Encourage the use of mixed-case, alphanumeric passwords, and strongly encourage the addition of at least one non-alphanumeric character to all passwords:

- t1Te-Bf,te
- Lb@lbhom

6.1.2.2.3. Memorable

A password is strong only if it can be remembered. However, being memorable and being easily guessed too often go together. Therefore, give your user community some tips on the creation of memorable passwords that cannot be easily guessed.

For example, take a favorite saying or phrase, and use the first letters of each word as the starting point in the creation of a new password. The result is memorable (because the phrase on which it is based is itself memorable), yet the result contains no words.

Note

Keep in mind that just using the first letters of each word in a phrase is not sufficient to make a strong password. Always be sure to increase the password's character set by including mixed-case alphanumeric characters and at least one special character as well.

6.1.2.3. Password Aging

If at all possible, implement password aging at your organization. Password aging is a feature (available in many operating systems) that sets limits on the time that a given password is considered valid. At the end of a password's lifetime, the user is prompted to enter a new password, which can then be used until, it too, expires.

The key question regarding password aging that many system administrators face is that of the password lifetime. What should it be?

There are two diametrically-opposed issues at work with respect to password lifetime:

- User convenience
- Security

On one extreme, a password lifetime of 99 years would present very little (if any) user inconvenience. However, it would provide very little (if any) security enhancement.

On the other extreme, a password lifetime of 99 minutes would be a large inconvenience to your users. However, security would be greatly enhanced.

The idea is to find a balance between your users' desired for convenience and your organization's need for security. For most organizations, password lifetimes in the weeks-to-months range are most common.

6.1.3. Access Control Information

Along with a username and password, user accounts also contain access control information. This information takes on different forms according to the operating system being used. However, the types of information often include:

- System-wide user-specific identification
- System-wide group-specific identification
- Lists of additional groups/capabilities to which the user is a member
- Default access information to be applied to all user-created files and resources

In some organizations, a user's access control information may never need to be touched. This is most often the case with standalone, personal workstations, for example. Other organizations, particularly those that make extensive use of network-wide resource sharing among different groups of users, require that a user's access control information be extensively modified.

The workload required to properly maintain your users' access control information varies according to how extensively your organization uses your operating system's access control features. While it is not a bad thing to rely so heavily on these features (in fact, it may be unavoidable), it does mean that your system environment may require more effort to maintain, and that every user account can have more ways in which it can be mis-configured.

Therefore, if your organization requires this kind of environment, you should make a point of documenting the exact steps required to create and correctly configure a user account. In fact, if there are different types of user accounts, you should document each one (creating a new finance user account, a new operations user account, etc.).

6.1.4. Managing Accounts and Resource Access Day-to-Day

As the old saying goes, the only constant is change. It is no different when dealing with your user community. People come, people go, and people move from one set of responsibilities to another. Therefore, system administrators must be able to respond to the changes that are a normal part of day-to-day life in your organization.

6.1.4.1. New Hires

When a new person joins your organization, they are normally given access to various resources (depending on their responsibilities). They may be given a place to work, a phone, and a key to the front door.

They may also be given access to one or more of the computers in your organization. As a system administrator, it is your responsibility to see that this is done promptly and appropriately. How should you do this?

Before you can do anything, you must first be aware of the new person's arrival. This is handled differently in various organizations. Here are some possibilities:

- Create a procedure where your organization's personnel department notifies you when a new person arrives.
- Create a form that the person's supervisor can fill out and use to request an account for the new person.

Different organizations require different approaches. However it is done, it is vital that you have a highly-reliable process that can alert you to any account-related work that needs to be done.

6.1.4.2. Terminations

The fact that people will be leaving your organization is a given. Sometimes it may be under happy circumstances and sometimes it may be under unhappy circumstances. In either case, it is vital that you are made aware of the situation so that you can take the appropriate actions.

At the very least, the appropriate actions should include:

- Disabling the user's access to all systems and related resources (usually by changing/locking the user's password)
- Backing up the user's files, in case they contain something that is needed at a later time
- Coordinating access to the user's files by the user's manager

The top priority is to secure your systems against the newly-terminated user. This is particularly important if the user was terminated under conditions that could leave the user feeling malice toward your organization. However, even if the circumstances are not quite so dire, it is in your organization's best interest for you to quickly and reliably disable access by the newly-terminated person.

This indicates the need for a process that alerts you to all terminations — preferably even before the actual termination takes place. This implies that you should work with your organization's personnel department to ensure that you are alerted to any upcoming terminations.

Note

When handling system "lock-downs" in response to terminations, proper timing is important. If the lock-down takes place after the termination process has been completed, there is the potential for unauthorized access by the newly-terminated person. On the other hand, if the lock-down takes place before the termination process has been initiated, it could alert the person to their impending termination, and make the process more difficult for all parties.

The termination process is usually initiated by a meeting between the person to be terminated, the person's manager, and a representative of your organization's personnel department. Therefore, putting a process in place that alerts you to the termination as this meeting starts ensures that the timing of the lock-down is appropriate.

Once access has been disabled, it is then time to make a backup copy of the newly-terminated person's files. This backup may be part of your organization's standard backups, or it may be a backup procedure dedicated to backing up old user accounts. Issues such as data retention regulations, preserving evidence in case of a wrongful termination lawsuit, and the like all play a part in determining the most appropriate way to handle backups.

In any case, a backup at this point is a good practice, as the next step (manager access to the newly-terminated person's files) may result in accidentally-deleted files. In such circumstances, having a current backup makes it possible to easily recover from any such accidents, making the process easier on the manager and you.

At this point, you must determine what access the newly-terminated person's manager requires to the person's files. Depending on your organization and the nature of the person's responsibilities, it might be that no access is required, or that access to everything will be necessary.

If the person used your systems for more than incidental email, it is likely that the manager has to sift through the files, determine what must be kept, and what may be discarded. As this process concludes, at least some of the files may be given to the person or persons taking over the newly-terminated person's responsibilities. Your assistance may be required in this final step of the process, or the manager may be in a position to handle this themselves. It all depends on the files and the nature of the work your organization undertakes.

6.1.4.3. Job Changes

Responding to requests to create accounts for new users and handling the sequence of events necessary to lock-down an account when a person is terminated are both relatively straightforward processes. However, it is not so clear-cut when a person changes responsibilities within your organization. Sometimes the person may require changes to their accounts and sometimes they may not.

There will be at least three people involved in making sure the user's account is appropriately reconfigured to match their new responsibilities:

- You
- The user's original manager
- The user's new manager

Between the three of you, it should be possible to determine what must take place to cleanly close out the user's old responsibilities, and what must be done to prepare the user's account for their new responsibilities. In many ways, this process can be thought of as being equivalent to shutting down an existing user account and creating a new user account. In fact, some organizations do this for all changes in responsibility.

However, it is more likely that the user's account will be kept and modified as appropriate to support their new responsibilities. This approach means that you must carefully review the account to ensure that it has only those resources and privileges appropriate to the person's new responsibilities.

Further complicating the situation is the fact that often there is a transition period where the user performs tasks related to both sets of responsibilities. This is where the user's original and new manager can help you by giving you a time frame for this transition period.

6.2. Managing User Resources

Creating user accounts is only part of a system administrator's job. Management of user resources is also essential. Therefore, three points must be considered:

- Who can access shared data
- Where users access this data
- What barriers are in place to prevent abuse of resources

The following sections briefly review each of these topics.

6.2.1. Who Can Access Shared Data

A user's access to a given application, file, or directory is determined by the permissions applied to that application, file, or directory.

In addition, it is often helpful if different permissions can be applied to different classes of users. For example, shared temporary storage should be capable of preventing the accidental (or malicious) deletions of a user's files by all other users, while still permitting the file's owner full access.

Another example is the access assigned to a user's home directory. Only the owner of the home directory should be able to create or view files there. Other users should be denied all access (unless the user wishes otherwise). This increases user privacy and prevents possible misappropriation of personal files.

But there are many situations where multiple users may need access to the same resources on a machine. In this case, careful creation of shared groups may be necessary.

6.2.1.1. Shared Groups and Data

As mentioned in the introduction, groups are logical constructs that can be used to cluster user accounts together for a specific purpose.

When managing users within an organization, it is wise to identify what data should be accessed by certain departments, what data should be denied to others, and what data should be shared by all. Determining this aids in the creation of an appropriate group structure, along with permissions appropriate for the shared data.

For instance, assume that that the accounts receivable department must maintain a list of accounts that are delinquent on their payments. They must also share that list with the collections department. If both accounts receivable and collections personnel are made members of a group called **`accounts`**, this information can then be placed in a shared directory (owned by the **`accounts`** group) with group read and write permissions on the directory.

6.2.1.2. Determining Group Structure

Some of the challenges facing system administrators when creating shared groups are:

- What groups to create
- Who to put in a given group
- What type of permissions should these shared resources have

A common-sense approach to these questions is helpful. One possibility is to mirror your organization's structure when creating groups. For example, if there is a finance department, create a group called **`finance`**, and make all finance personnel members of that group. If the financial information is too sensitive for the company at large, but vital for senior officials within the organization, then grant the senior officials group-level permission to access the directories and data used by the finance department by adding all senior officials to the **`finance`** group.

It is also good to be cautious when granting permissions to users. This way, sensitive information is less likely to fall into the wrong hands.

By approaching the creation of your organization's group structure in this manner, the need for access to shared data within the organization can be safely and effectively met.

6.2.2. Where Users Access Shared Data

When sharing data among users, it is a common practice to have a central server (or group of servers) that make certain directories available to other machines on the network. This way data is stored in one place; synchronizing data between multiple machines is not necessary.

Before taking this approach, you must first determine what systems are to access the centrally-stored data. As you do this, take note of the operating systems used by the systems. This information has a bearing on your ability to implement such an approach, as your storage server must be capable of serving its data to each of the operating systems in use at your organization.

Unfortunately, once data is shared between multiple computers on a network, the potential for conflicts in file ownership can arise.

6.2.2.1. Global Ownership Issues

There are benefits if data is stored centrally and is accessed by multiple computers over a network. However, assume for a moment that each of those computers has a locally-maintained list of user accounts. What if the list of users on each of these systems are not consistent with the list of users on the central server? Even worse, what if the list of users on each of these systems are not even consistent with each other?

Much of this depends on how users and access permissions are implemented on each system, but in some cases it is possible that user A on one system may actually be known as user B on another system. This becomes a real problem when data is shared between these systems, as data that user A is allowed to access from one system can also be read by user B from another system.

For this reason, many organizations use some sort of central user database. This assures that there are no overlaps between user lists on different systems.

6.2.2.2. Home Directories

Another issue facing system administrators is whether or not users should have centrally-stored home directories.

The primary advantage of centralizing home directories on a network-attached server is that if a user logs into any machine on the network, they will be able to access the files in their home directory.

The disadvantage is that if the network goes down, users across the entire organization will be unable to get to their files. In some situations (such as organizations that make widespread use of laptops), having centralized home directories may not be desirable. But if it makes sense for your organization, deploying centralized home directories can make a system administrator's life much easier.

6.2.3. What Barriers Are in Place To Prevent Abuse of Resources

The careful organization of groups and assignment of permissions for shared resources is one of the most important things a system administrator can do to prevent resource abuse among users within an organization. In this way, those who should not have access to sensitive resources are denied access.

But no matter how your organization does things, the best guard against abuse of resources is always sustained vigilance on the part of the system administrator. Keeping your eyes open is often the only way to avoid having an unpleasant surprise waiting for you at your desk one morning.

6.3. Red Hat Enterprise Linux-Specific Information

The following sections describe the various features specific to Red Hat Enterprise Linux that relate to the administration of user accounts and associated resources.

6.3.1. User Accounts, Groups, and Permissions

Under Red Hat Enterprise Linux, a user can log into the system and use any applications or files they are permitted to access after a normal user account is created. Red Hat Enterprise Linux determines whether or not a user or group can access these resources based on the permissions assigned to them.

There are three different permissions for files, directories, and applications. These permissions are used to control the kinds of access allowed. Different one-character symbols are used to describe each permission in a directory listing. The following symbols are used:

- **`r`** — Indicates that a given category of user can read a file.
- **`w`** — Indicates that a given category of user can write to a file.
- **`x`** — Indicates that a given category of user can execute the contents of a file.

A fourth symbol (**`-`**) indicates that no access is permitted.

Each of the three permissions are assigned to three different categories of users. The categories are:

- *owner* — The owner of the file or application.
- *group* — The group that owns the file or application.
- *everyone* — All users with access to the system.

As stated earlier, it is possible to view the permissions for a file by invoking a long format listing with the command **`ls -l`**. For example, if the user **`juan`** creates an executable file named **`foo`**, the output of the command **`ls -l foo`** would appear like this:

```
-rwxrwxr-x 1 juan juan 0 Sep 26 12:25 foo
```

The permissions for this file are listed at the start of the line, beginning with **`rwx`**. This first set of symbols define owner access — in this example, the owner **`juan`** has full access, and may read, write, and execute the file. The next set of **`rwx`** symbols define group access (again, with full access), while the last set of symbols define the types of access permitted for all other users. Here, all other users may read and execute the file, but may not modify it in any way.

One important point to keep in mind regarding permissions and user accounts is that every application run on Red Hat Enterprise Linux runs in the context of a specific user. Typically, this means that if user **`juan`** launches an application, the application runs using user **`juan`**'s context. However, in some cases the application may need a more privileged level of access in order to accomplish a task. Such applications include those that edit system settings or log in users. For this reason, special permissions have been created.

There are three such special permissions within Red Hat Enterprise Linux. They are:

- *setuid* — used only for binary files (applications), this permission indicates that the file is to be executed with the permissions of the owner of the file, and not with the permissions of the user executing the file (which is the case without setuid). This is indicated by the character **`s`** in the place of the **`x`** in the owner category. If the owner of the file does not have execute permissions, a capital **`S`** reflects this fact.

- *setgid* — used primarily for binary files (applications), this permission indicates that the file is executed with the permissions of the group owning the file and not with the permissions of the group of the user executing the file (which is the case without setgid).

  If applied to a directory, all files created within the directory are owned by the group owning the directory, and not by the group of the user creating the file. The setgid permission is indicated by the character **`s`** in place of the **`x`** in the group category. If the group owning the file or directory does not have execute permissions, a capital **`S`** reflects this fact.

- *sticky bit* — used primarily on directories, this bit dictates that a file created in the directory can be removed only by the user that created the file. It is indicated by the character **`t`** in place of the **`x`** in the everyone category. If the everyone category does not have execute permissions, the **`T`** is capitalized to reflect this fact.

  Under Red Hat Enterprise Linux, the sticky bit is set by default on the **`/tmp/`** directory for exactly this reason.

6.3.1.1. Usernames and UIDs, Groups and GIDs

In Red Hat Enterprise Linux, user account and group names are primarily for peoples' convenience. Internally, the system uses numeric identifiers. For users, this identifier is known as a *UID*, while for groups the identifier is known as a *GID*. Programs that make user or group information available to users translate the UID/GID values into their more human-readable counterparts.

Important

UIDs and GIDs must be globally unique within your organization if you intend to share files and resources over a network. Otherwise, whatever access controls you put in place may fail to work properly, as they are based on UIDs and GIDs, not usernames and group names.

Specifically, if the **`/etc/passwd`** and **`/etc/group`** files on a file server and a user's workstation differ in the UIDs or GIDs they contain, improper application of permissions can lead to security issues.

For example, if user **`juan`** has a UID of 500 on a desktop computer, files **`juan`** creates on a file server will be created with owner UID 500. However, if user **bob** logs in locally to the file server (or even some other computer), and **bob**'s account also has a UID of 500, **bob** will have full access to **`juan`**'s files, and vice versa.

Therefore, UID and GID collisions are to be avoided at all costs.

There are two instances where the actual numeric value of a UID or GID has any specific meaning. A UID and GID of zero (0) are used for the **`root`** user, and are treated specially by Red Hat Enterprise Linux — all access is automatically granted.

The second instance is that UIDs and GIDs below 500 are reserved for system use. Unlike UID/GID zero (0), UIDs and GIDs below 500 are not treated specially by Red Hat Enterprise Linux. However, these UIDs/GIDs are never to be assigned to a user, as it is likely that some system component either currently uses or will use these UIDs/GIDs at some point in the future. For more information on these standard users and groups, see the chapter titled *Users and Groups* in the *Reference Guide*.

When new user accounts are added using the standard Red Hat Enterprise Linux user creation tools, the new user accounts are assigned the first available UID and GID starting at 500. The next new user account is assigned UID/GID 501, followed by UID/GID 502, and so on.

A brief overview of the various user creation tools available under Red Hat Enterprise Linux occurs later in this chapter. But before reviewing these tools, the next section reviews the files Red Hat Enterprise Linux uses to define system accounts and groups.

6.3.2. Files Controlling User Accounts and Groups

On Red Hat Enterprise Linux, information about user accounts and groups are stored in several text files within the **`/etc/`** directory. When a system administrator creates new user accounts, these files must either be edited manually or applications must be used to make the necessary changes.

The following section documents the files in the **`/etc/`** directory that store user and group information under Red Hat Enterprise Linux.

6.3.2.1. `/etc/passwd`

The **`/etc/passwd`** file is world-readable and contains a list of users, each on a separate line. On each line is a colon delimited list containing the following information:

- *Username* — The name the user types when logging into the system.
- *Password* — Contains the encrypted password (or an **`x`** if shadow passwords are being used — more on this later).
- *User ID* (*UID*) — The numerical equivalent of the username which is referenced by the system and applications when determining access privileges.

- *Group ID* (*GID*) — The numerical equivalent of the primary group name which is referenced by the system and applications when determining access privileges.
- *GECOS* — Named for historical reasons, the GECOS[1] field is optional and is used to store extra information (such as the user's full name). Multiple entries can be stored here in a comma delimited list. Utilities such as **finger** access this field to provide additional user information.
- *Home directory* — The absolute path to the user's home directory, such as **/home/juan/**.
- *Shell* — The program automatically launched whenever a user logs in. This is usually a command interpreter (often called a *shell*). Under Red Hat Enterprise Linux, the default value is **/bin/bash**. If this field is left blank, **/bin/sh** is used. If it is set to a non-existent file, then the user will be unable to log into the system.

Here is an example of a **/etc/passwd** entry:

```
root:x:0:0:root:/root:/bin/bash
```

This line shows that the **root** user has a shadow password, as well as a UID and GID of 0. The **root** user has **/root/** as a home directory, and uses **/bin/bash** for a shell.

For more information about **/etc/passwd**, see the **passwd(5)** man page.

6.3.2.2. /etc/shadow

Because the **/etc/passwd** file must be world-readable (the main reason being that this file is used to perform the translation from UID to username), there is a risk involved in storing everyone's password in **/etc/passwd**. True, the passwords are encrypted. However, it is possible to perform attacks against passwords if the encrypted password is available.

If a copy of **/etc/passwd** can be obtained by an attacker, an attack that can be carried out in secret becomes possible. Instead of risking detection by having to attempt an actual login with every potential password generated by password-cracker, an attacker can use a password cracker in the following manner:

- A password-cracker generates potential passwords
- Each potential password is then encrypted using the same algorithm as the system
- The encrypted potential password is then compared against the encrypted passwords in **/etc/passwd**

The most dangerous aspect of this attack is that it can take place on a system far-removed from your organization. Because of this, the attacker can use the highest-performance hardware available, making it possible to go through massive numbers of passwords very quickly.

Therefore, the **/etc/shadow** file is readable only by the root user and contains password (and optional password aging information) for each user. As in the **/etc/passwd** file, each user's information is on a separate line. Each of these lines is a colon delimited list including the following information:

- *Username* — The name the user types when logging into the system. This allows the **login** application to retrieve the user's password (and related information).
- *Encrypted password* — The 13 to 24 character password. The password is encrypted using either the **crypt(3)** library function or the md5 hash algorithm. In this field, values other than a validly-

formatted encrypted or hashed password are used to control user logins and to show the password status. For example, if the value is **`!`** or **`*`**, the account is locked and the user is not allowed to log in. If the value is **`!!`** a password has never been set before (and the user, not having set a password, will not be able to log in).

- *Date password last changed* — The number of days since January 1, 1970 (also called the *epoch*) that the password was last changed. This information is used in conjunction with the password aging fields that follow.
- *Number of days before password can be changed* — The minimum number of days that must pass before the password can be changed.
- *Number of days before a password change is required* — The number of days that must pass before the password must be changed.
- *Number of days warning before password change* — The number of days before password expiration during which the user is warned of the impending expiration.
- *Number of days before the account is disabled* — The number of days after a password expires before the account will be disabled.
- *Date since the account has been disabled* — The date (stored as the number of days since the epoch) since the user account has been disabled.
- *A reserved field* — A field that is ignored in Red Hat Enterprise Linux.

Here is an example line from **`/etc/shadow`**:

```
juan:$1$.QKDPc5E$SWlkjRWexrXYgc98F.:12825:0:90:5:30:13096:
```

This line shows the following information for user **`juan`**:

- The password was last changed February 11, 2005
- There is no minimum amount of time required before the password can be changed
- The password must be changed every 90 days
- The user will get a warning five days before the password must be changed
- The account will be disabled 30 days after the password expires if no login attempt is made
- The account will expire on November 9,2005

For more information on the **`/etc/shadow`** file, see the **`shadow(5)`** man page.

6.3.2.3. `/etc/group`

The **`/etc/group`** file is world-readable and contains a list of groups, each on a separate line. Each line is a four field, colon delimited list including the following information:

- *Group name* — The name of the group. Used by various utility programs as a human-readable identifier for the group.
- *Group password* — If set, this allows users that are not part of the group to join the group by using the **`newgrp`** command and typing the password stored here. If a lower case **`x`** is in this field, then shadow group passwords are being used.

- *Group ID* (*GID*) — The numerical equivalent of the group name. It is used by the operating system and applications when determining access privileges.
- *Member list* — A comma delimited list of the users belonging to the group.

Here is an example line from **/etc/group**:

```
general:x:502:juan,shelley,bob
```

This line shows that the **general** group is using shadow passwords, has a GID of 502, and that **juan**, **shelley**, and **bob** are members.

For more information on **/etc/group**, see the **group(5)** man page.

6.3.2.4. /etc/gshadow

The **/etc/gshadow** file is readable only by the root user and contains an encrypted password for each group, as well as group membership and administrator information. Just as in the **/etc/group** file, each group's information is on a separate line. Each of these lines is a colon delimited list including the following information:

- *Group name* — The name of the group. Used by various utility programs as a human-readable identifier for the group.
- *Encrypted password* — The encrypted password for the group. If set, non-members of the group can join the group by typing the password for that group using the **newgrp** command. If the value of this field is **!**, then no user is allowed to access the group using the **newgrp** command. A value of **!!** is treated the same as a value of **!** — however, it also indicates that a password has never been set before. If the value is null, only group members can log into the group.
- *Group administrators* — Group members listed here (in a comma delimited list) can add or remove group members using the **gpasswd** command.
- *Group members* — Group members listed here (in a comma delimited list) are regular, non-administrative members of the group.

Here is an example line from **/etc/gshadow**:

```
general:!!:shelley:juan,bob
```

This line shows that the **general** group has no password and does not allow non-members to join using the **newgrp** command. In addition, **shelley** is a group administrator, and **juan** and **bob** are regular, non-administrative members.

Since editing these files manually raises the potential for syntax errors, it is recommended that the applications provided with Red Hat Enterprise Linux for this purpose be used instead. The next section reviews the primary tools for performing these tasks.

6.3.3. User Account and Group Applications

There are two basic types of applications that can be used when managing user accounts and groups on Red Hat Enterprise Linux systems:

- The graphical **User Management Tool** application

- A suite of command line tools

For detailed instructions on using **User Management Tool**, see the chapter titled *User and Group Configuration* in the *System Administrators Guide*.

While both the **User Management Tool** application and the command line utilities perform essentially the same task, the command line tools have the advantage of being script-able and are therefore more easily automated.

The following table describes some of the more common command line tools used to create and manage user accounts and groups:

Table 6.2. User Management Command Line Tools

Application	Function
`/usr/sbin/useradd`	Adds user accounts. This tool is also used to specify primary and secondary group membership.
`/usr/sbin/userdel`	Deletes user accounts.
`/usr/sbin/usermod`	Edits account attributes including some functions related to password aging. For more fine-grained control, use the **`passwd`** command. **`usermod`** is also used to specify primary and secondary group membership.
`passwd`	Sets passwords. Although primarily used to change a user's password, it also controls all aspects of password aging.
`/usr/sbin/chpasswd`	Reads in a file consisting of username and password pairs, and updates each users' password accordingly.
`chage`	Changes the user's password aging policies. The **`passwd`** command can also be used for this purpose.
`chfn`	Changes the user's GECOS information.
`chsh`	Changes the user's default shell.

The following table describes some of the more common command line tools used to create and manage groups:

Table 6.3. Group Management Command Line Tools

Application	Function
`/usr/sbin/groupadd`	Adds groups, but does not assign users to those groups. The **`useradd`** and **`usermod`** programs should then be used to assign users to a given group.
`/usr/sbin/groupdel`	Deletes groups.
`/usr/sbin/groupmod`	Modifies group names or GIDs, but does not change group membership. The **`useradd`** and **`usermod`** programs should be used to assign users to a given group.
`gpasswd`	Changes group membership and sets passwords to allow non-group members who know the group password to join the group. It is also used to specify group administrators.
`/usr/sbin/grpck`	Checks the integrity of the **`/etc/group`** and **`/etc/gshadow`** files.

The tools listed thus far provide system administrators great flexibility in controlling all aspects of user accounts and group membership. To learn more about how they work, refer to the man page for each.

These applications do not, however, determine what resources these users and groups have control over. For this, the system administrator must use file permission applications.

6.3.3.1. File Permission Applications

File permissions are an integral part of managing resources within an organization. The following table describes some of the more common command line tools used for this purpose.

Table 6.4. Permission Management Command Line Tools

Application	Function
`chgrp`	Changes which group owns a given file.
`chmod`	Changes access permissions for a given file. It is also capable of assigning special permissions.
`chown`	Changes a file's ownership (and can also change group).

It is also possible to alter these attributes in the GNOME and KDE graphical environments. Right-click on the the file's icon (for example, while the icon is displayed in a graphical file manager or on the desktop), and select **Properties**.

6.4. Additional Resources

This section includes various resources that can be used to learn more about account and resource management, and the Red Hat Enterprise Linux-specific subject matter discussed in this chapter.

6.4.1. Installed Documentation

The following resources are installed in the course of a typical Red Hat Enterprise Linux installation and can help you learn more about the subject matter discussed in this chapter.

- **User Management ToolHelp** menu entry — Learn how to manage user accounts and groups.
- **`passwd(5)`** man page — Learn more about the file format information for the **`/etc/passwd`** file.
- **`group(5)`** man page — Learn more about the file format information for the **`/etc/group`** file.
- **`shadow(5)`** man page — Learn more about the file format information for the **`/etc/shadow`** file.
- **`useradd(8)`** man page — Learn how to create or update user accounts.
- **`userdel(8)`** man page — Learn how to delete user accounts.
- **`usermod(8)`** man page — Learn how to modify user accounts.
- **`passwd(1)`** man page — Learn how to update a user's password.
- **`chpasswd(8)`** man page — Learn how to update many users' passwords at one time.
- **`chage(1)`** man page — Learn how to change user password aging information.
- **`chfn(1)`** man page — Learn how to change a user's GECOS (**`finger`**) information.
- **`chsh(1)`** man page — Learn how to change a user's login shell.
- **`groupadd(8)`** man page — Learn how to create a new group.
- **`groupdel(8)`** man page — Learn how to delete a group.

- **`groupmod(8)`** man page — Learn how to modify a group.
- **`gpasswd(1)`** man page — Learn how to administer the **`/etc/group`** and **`/etc/gshadow`** files.
- **`grpck(1)`** man page — Learn how to verify the integrity of the **`/etc/group`** and **`/etc/gshadow`** files.
- **`chgrp(1)`** man page — Learn how to change group-level ownership.
- **`chmod(1)`** man page — Learn how to change file access permissions.
- **`chown(1)`** man page — Learn how to change file owner and group.

6.4.2. Useful Websites

- *http://www.bergen.org/ATC/Course/InfoTech/passwords.html* — A good example of a document conveying information about password security to an organization's users.
- *http://www.crypticide.org/users/alecm/* — Homepage of the author of one of the most popular password-cracking programs (Crack). You can download Crack from this page and see how many of your users have weak passwords.
- *http://www.linuxpowered.com/html/editorials/file.html* — a good overview of Linux file permissions.

6.4.3. Related Books

The following books discuss various issues related to account and resource management, and are good resources for Red Hat Enterprise Linux system administrators.

- The *Security Guide*; Red Hat, Inc — Provides an overview of the security-related aspects of user accounts, namely choosing strong passwords.
- The *Reference Guide*; Red Hat, Inc — Contains detailed information on the users and groups present in Red Hat Enterprise Linux.
- The *System Administrators Guide*; Red Hat, Inc — Includes a chapter on user and group configuration.
- *Linux Administration Handbook* by Evi Nemeth, Garth Snyder, and Trent R. Hein; Prentice Hall — Provides a chapter on user account maintenance, a section on security as it relates to user account files, and a section on file attributes and permissions.

Printers and Printing

Printers are an essential resource for creating a *hard copy* -- a physical depiction of data on paper -- version of documents and collateral for business, academic, and home use. Printers have become an indispensable peripheral in all levels of business and institutional computing.

This chapter discusses the various printers available and compares their uses in different computing environments. It then describes how printing is supported by Red Hat Enterprise Linux.

7.1. Types of Printers

Like any other computer peripheral, there are several types of printers available. Some printers employ technologies that mimic manual typewriter-style functionality, while others spray ink on paper, or use a laser to generate an image of the page to be printed. Printer hardware interfaces with a PC or network using parallel, serial, or data networking protocols. There are several factors to consider when evaluating printers for procurement and deployment in your computing environment.

The following sections discuss the various printer types and the protocols that printers use to communicate with computers.

7.1.1. Printing Considerations

There are several aspects to factor into printer evaluations. The following specifies some of the most common criteria when evaluating your printing needs.

7.1.1.1. Function

Evaluating your organizational needs and how a printer services those needs is the essential criteria in determining the right type of printer for your environment. The most important question to ask is "*What do we need to print?*" Since there are specialized printers for text, images, or any variation thereof, you should be certain that you procure the right tool for your purposes.

For example, if your requirements call for high-quality color images on professional-grade glossy paper, it is recommended you use a dye-sublimation or thermal wax transfer color printer instead of a laser or impact printer.

Conversely, laser or inkjet printers are well-suited for printing rough drafts or documents intended for internal distribution (such high-volume printers are usually called *workgroup* printers). Determining the needs of the everyday user allows administrators to determine the right printer for the job.

Other factors to consider are features such as *duplexing* -- the ability to print on both sides of a piece of paper. Traditionally, printers could only print on one side of the page (called *simplex* printing). Most lower-end printer models today do not have duplexing by default (they may, however, be capable of a manual duplexing method that requires the user to flip the paper themselves). Some models offer add-on hardware for duplexing; such add-ons can drive one-time costs up considerably. However, duplex printing may reduce costs over time by reducing the amount of paper used to print documents, thus reducing the cost of *consumables* -- primarily paper.

Another factor to consider is paper size. Most printers are capable of handling the more common paper sizes:

- *letter* -- (8 1/2" x 11")
- *A4* -- (210mm x 297mm)
- *JIS B5* -- (182mm x 257mm)

- *legal* -- (8 1/2" x 14")

If certain departments (such as marketing or design) have specialized needs such as creating posters or banners, there are *large-format* printers capable of using *A3* (297mm x 420mm) or tabloid (11" x 17") paper sizes. In addition, there are printers capable of even larger sizes, although these are often only used for specialized purposes, such as printing blueprints.

Additionally, high-end features such as network modules for workgroup and remote site printing should also be considered during evaluation.

7.1.1.2. Cost

Cost is another factor to consider when evaluating printers. However, determining the one-time cost associated with the purchase of the printer itself is not sufficient. There are other costs to consider, such as consumables, parts and maintenance, and printer add-ons.

As the name implies, consumables is a general term used to describe the material used up during the printing process. Consumables primarily take the form of *media* and *ink*.

The media is the material on which the text or image is printed. The choice of media is heavily dependent on the type of information being printed.

For example, creating an accurate print of a digital image requires a special glossy paper that can withstand prolonged exposure to natural or artificial lighting, as well as ensure accuracy of color reproduction; these qualities are known as color *fastness*. For *archival-quality* documents that require durability and a professional level of legibility (such as contracts, résumés, and permanent records), a *matte* (or non-glossy) paper should be used. The *stock* (or thickness) of paper is also important, as some printers have a paper path that is not straight. The use of paper that is too thin or too thick can result in jams. Some printers can also print on *transparencies*, allowing the information to be projected on a screen during presentations.

Specialized media such as those noted here can affect the cost of consumables, and should be taken into consideration when evaluating printing needs.

Ink is a generalized term, as not all printers use liquid inks. For example, laser printers use a powder known as *toner*, while impact printers use ribbons saturated with ink. There are specialized printers that heat the ink during the printing process, while others spray small droplets of ink onto the media. Ink replacement costs vary widely and depend on whether the container holding the ink can be *recharged* (refilled) or if it requires a complete replacement of the ink *cartridge*.

7.2. Impact Printers

Impact printers are the oldest printing technologies still in active production. Some of the largest printer vendors continue to manufacture, market, and support impact printers, parts, and supplies. Impact printers are most functional in specialized environments where low-cost printing is essential. The three most common forms of impact printers are *dot-matrix*, *daisy-wheel*, and *line printers*.

7.2.1. Dot-Matrix Printers

The technology behind dot-matrix printing is quite simple. The paper is pressed against a *drum* (a rubber-coated cylinder) and is intermittently pulled forward as printing progresses. The electromagnetically-driven *printhead* moves across the paper and strikes the printer ribbon situated between the paper and printhead pin. The impact of the printhead against the printer ribbon imprints ink dots on the paper which form human-readable characters.

Dot-matrix printers vary in print resolution and overall quality with either 9 or 24-pin printheads. The more pins per inch, the higher the print resolution. Most dot-matrix printers have a maximum resolution of around 240 *dpi* (dots per inch). While this resolution is not as high as those possible in laser or inkjet printers, there is one distinct advantage to dot-matrix (or any form of impact) printing. Because the printhead must strike the surface of the paper with enough force to transfer ink from a ribbon onto the page, it is ideal for environments that must produce *carbon copies* through the use of special multi-part documents. These documents have carbon (or other pressure-sensitive material) on the underside and create a mark on the sheet underneath when pressure is applied. Retailers and small businesses often use carbon copies as receipts or bills of sale.

7.2.2. Daisy-Wheel Printers

If you have ever worked with a manual typewriter before, then you understand the technological concept behind daisy-wheel printers. These printers have printheads composed of metallic or plastic wheels cut into *petals*. Each petal has the form of a letter (in capital and lower-case), number, or punctuation mark on it. When the petal is struck against the printer ribbon, the resulting shape forces ink onto the paper. Daisy-wheel printers are loud and slow. They cannot print graphics, and cannot change fonts unless the print wheel is physically replaced. With the advent of laser printers, daisy-wheel printers are generally not used in modern computing environments.

7.2.3. Line Printers

Another type of impact printer somewhat similar to the daisy-wheel is the *line printer*. However, instead of a print wheel, line printers have a mechanism that allows multiple characters to be simultaneously printed on the same line. The mechanism may use a large spinning *print drum* or a looped *print chain*. As the drum or chain is rotated over the paper's surface, electromechanical hammers behind the paper push the paper (along with a ribbon) onto the surface of the drum or chain, marking the paper with the shape of the character on the drum or chain.

Because of the nature of the print mechanism, line printers are much faster than dot-matrix or daisy-wheel printers. However, they tend to be quite loud, have limited multi-font capability, and often produce lower print quality than more recent printing technologies.

Because line printers are used for their speed, they use special *tractor-fed* paper with pre-punched holes along each side. This arrangement makes continuous unattended high-speed printing possible, with stops only required when a box of paper runs out.

7.2.4. Impact Printer Consumables

Of all the printer types, impact printers have relatively low consumable costs. Ink ribbons and paper are the primary recurring costs for impact printers. Some Impact printers (usually line and dot-matrix printers) require tractor-fed paper, which can increase the costs of operation somewhat.

7.3. Inkjet Printers

An *Inkjet* printer uses one of the most popular printing technologies today. The relatively low cost and multi-purpose printing abilities make inkjet printers a good choice for small businesses and home offices.

Inkjet printers use quick-drying, water-based inks and a printhead with a series of small nozzles that spray ink onto the surface of the paper. The printhead assembly is driven by a belt-fed motor that moves the printhead across the paper.

Inkjets were originally manufactured to print in *monochrome* (black and white) only. However, the printhead has since been expanded and the nozzles increased to accommodate cyan, magenta, yellow, and black. This combination of colors (called *CMYK*) allows the printing of images with nearly the same quality as a photo development lab (when using certain types of coated paper.) When coupled with crisp and highly readable text print quality, inkjet printers are a sound all-in-one choice for monochrome or color printing needs.

7.3.1. Inkjet Consumables

Inkjet printers tend to be low cost and scale slightly upward based on print quality, extra features, and the ability to print on larger formats than the standard legal or letter paper sizes. While the one-time cost of purchasing an inkjet printer is lower than other printer types, there is the factor of inkjet consumables that must be considered. Because demand for inkjets is large and spans the computing spectrum from home to enterprise, the procurement of consumables can be costly.

Note

When shopping for an inkjet printer, always make sure you know what kind of ink cartridge(s) it requires. This is especially critical for color units. CMYK inkjet printers require ink for each color; however, the important point is whether each color is stored in a separate cartridge or not.

Some printers use one multi-chambered cartridge; unless some sort of refilling process is possible, as soon as one color ink runs out, the entire cartridge must be replaced. Other printers use a multi-chambered cartridge for cyan, magenta, and yellow, but also have a separate cartridge for black. In environments where a great deal of text is printed, this type of arrangement can be beneficial. However, the best solution is to find a printer with separate cartridges for each color; you can then easily replace any color whenever it runs out.

Some inkjet manufacturers also require you to use specially treated paper for printing high-quality images and documents. Such paper uses a moderate to high gloss coating formulated to absorb colored inks, which prevents *clumping* (the tendency for water-based inks to collect in certain areas where colors blend, causing muddiness or dried ink blots) or *banding* (where the print output has a striped pattern of extraneous lines on the printed page.) Consult your printer's documentation for recommended papers.

7.4. Laser Printers

An older technology than inkjet, laser printers are another popular alternative to legacy impact printing. Laser printers are known for their high volume output and low cost-per-page. Laser printers are often deployed in enterprises as a workgroup or departmental print center, where performance, durability, and output requirements are a priority. Because laser printers service these needs so readily (and at a reasonable cost-per-page), the technology is widely regarded as the workhorse of enterprise printing.

Laser printers share much of the same technologies as photocopiers. Rollers pull a sheet of paper from a paper tray and through a *charge roller*, which gives the paper an electrostatic charge. At the same time, a printing drum is given the opposite charge. The surface of the drum is then scanned by a laser, discharging the drum's surface and leaving only those points corresponding to the desired text and image with a charge. This charge is then used to force toner to adhere to the drum's surface.

The paper and drum are then brought into contact; their differing charges cause the toner to then adhere to the paper. Finally, the paper travels between *fusing rollers*, which heat the paper and melt the toner, fusing it onto the paper's surface.

7.4.1. Color Laser Printers

Color laser printers aim to combine the best features of laser and inkjet technology into a multi-purpose printer package. The technology is based on traditional monochrome laser printing, but uses additional components to create color images and documents. Instead of using black toner only, color laser printers use a CMYK toner combination. The print drum either rotates each color and lays the toner down one color at a time, or lays all four colors down onto a plate and then passes the paper through the drum, transferring the complete image onto the paper. Color laser printers also employ *fuser oil* along with the heated fusing rolls, which further bonds the color toner to the paper and can give varying degrees of gloss to the finished image.

Because of their increased features, color laser printers are typically twice (or several times) as expensive as monochrome laser printers. In calculating the total cost of ownership with respect to printing resources, some administrators may wish to separate monochrome (text) and color (image) functionality to a dedicated monochrome laser printer and a dedicated color laser (or inkjet) printer, respectively.

7.4.2. Laser Printer Consumables

Depending on the type of laser printer deployed, consumable costs are usually proportional to the volume of printing. Toner comes in cartridges that are usually replaced outright; however, some models come with refillable cartridges. Color laser printers require one toner cartridge for each of the four colors. Additionally, color laser printers require fuser oils to bond toner onto paper and waste toner bottles to capture toner spillover. These added supplies raise the consumables cost of color laser printers; however, it is worth noting that such consumables, on average, last about 6000 pages, which is much greater than comparable inkjet or impact consumable lifespans. Paper type is less of an issue in laser printers, which means bulk purchases of regular xerographic or photocopy paper are acceptable for most print jobs. However, if you plan to print high-quality images, you should opt for glossy paper for a professional finish.

7.5. Other Printer Types

There are other types of printers available, mostly special-purpose printers for professional graphics or publishing organizations. These printers are not for general purpose use, however. Because they are relegated to niche uses, their prices (both one-time and recurring consumables costs) tend to be higher relative to more mainstream units.

Thermal Wax Printers

> These printers are used mostly for business presentation transparencies and for color *proofing* (creating test documents and images for close quality inspection before sending off master documents to be printed on industrial four-color offset printers). Thermal wax printers use sheet-sized, belt driven CMYK ribbons and specially-coated paper or transparencies. The printhead contains heating elements that melt each wax color onto the paper as it is rolled through the printer.

Dye-Sublimation Printers

> Used in organizations such as service bureaus -- where professional quality documents, pamphlets, and presentations are more important than consumables costs -- dye-sublimation (or dye-sub) printers are the workhorses of quality CMYK printing. The concepts behind dye-sub printers are similar to thermal wax printers except for the use of diffusive plastic dye film instead of colored wax. The printhead heats the colored film and vaporizes the image onto specially coated paper.
>
> Dye-sub is quite popular in the design and publishing world as well as the scientific research field, where preciseness and detail are required. Such detail and print quality comes at a price, as dye-sub printers are also known for their high costs-per-page.

Solid Ink Printers

Used mostly in the packaging and industrial design industries, solid ink printers are prized for their ability to print on a wide variety of paper types. Solid ink printers, as the name implies, use hardened ink sticks that that are melted and sprayed through small nozzles on the printhead. The paper is then sent through a fuser roller which further forces the ink onto the paper.

The solid ink printer is ideal for prototyping and proofing new designs for product packages; as such, most service-oriented businesses would not have a need for this type of printer.

7.6. Printer Languages and Technologies

Before the advent of laser and inkjet technology, impact printers could only print standard, justified text with no variation in letter size or font style. Today, printers are able to process complex documents with embedded images, charts, and tables in multiple frames and in several languages, all on one page. Such complexity must adhere to some format conventions. This is what spurred the development of the *page description language* (or PDL) -- a specialized document formatting language specially made for computer communication with printers.

Over the years, printer manufacturers have developed their own proprietary languages to describe document formats. However, such proprietary languages applied only to the printers that the manufacturers created themselves. If, for example, you were to send a print-ready file using a proprietary PDL to a professional press, there was no guarantee that your file would be compatible with the printer's machines. The issue of portability came into question.

Xerox® developed the Interpress™ protocol for their line of printers, but full adoption of the language by the rest of the printing industry was never realized. Two original developers of Interpress left Xerox and formed Adobe®, a software company catering mostly to electronic graphics and document professionals. At Adobe, they developed a widely-adopted PDL called *PostScript*™, which uses a markup language to describe text formatting and image information that could be processed by printers. At the same time, the Hewlett-Packard® Company developed the *Printer Control Language*™ (or PCL) for use in their ubiquitous laser and inkjet printer lines. PostScript and PCL are now widely adopted PDLs and are supported by most printer manufacturers.

PDLs work on the same principle as computer programming languages. When a document is ready for printing, the PC or workstation takes the images, typographical information, and document layout, and uses them as objects that form instructions for the printer to process. The printer then translates those objects into *rasters*, a series of scanned lines that form an image of the document (called *Raster Image Processing* or RIP), and prints the output onto the page as one image, complete with text and any graphics included. This process makes printed documents more consistent, resulting in little or no variation when printing the same document on different model printers. PDLs are designed to be portable to any format, and scalable to fit different paper sizes.

Choosing the right printer is a matter of determining what standards the various departments in your organization have adopted for their needs. Most departments use word processing and other productivity software that use the PostScript language for outputting to printers. However, if your graphics department requires PCL or some proprietary form of printing, you must take that into consideration as well.

7.7. Networked Versus Local Printers

Depending on organizational needs, it may be unnecessary to assign one printer to each member of your organization. Such overlap in expenditure can eat into allotted budgets, leaving less capital for other necessities. While local printers attached via a parallel or USB cable to every workstation are an ideal solution for the user, it is usually not economically feasible.

Printer manufacturers have addressed this need by developing *departmental* (or workgroup) printers. These machines are usually durable, fast, and have long-life consumables. Workgroup printers usually are attached to a print server, a standalone device (such as a reconfigured workstation) that handles print jobs and routes output to the proper printer when available. More recent departmental printers include built-in or add-on network interfaces that eliminate the need for a dedicated print server.

7.8. Red Hat Enterprise Linux-Specific Information

The following describes the various features specific to Red Hat Enterprise Linux that relate to printers and printing.

The **Printer Configuration Tool** allows users to configure a printer. This tool helps maintain the printer configuration file, print spool directories, and print filters.

Red Hat Enterprise Linux 4 uses the CUPS printing system. If a system was upgraded from a previous Red Hat Enterprise Linux version that used CUPS, the upgrade process preserved the configured queues.

Using the **Printer Configuration Tool** requires root privileges. To start the application, select **Main Menu Button** (on the Panel) => **System Settings** => **Printing**, or type the command **`system-config-printer`**. This command automatically determines whether to run the graphical or text-based version depending on whether the command is executed in the graphical desktop environment or from a text-based console.

To force the **Printer Configuration Tool** to run as a text-based application, execute the command **`system-config-printer-tui`** from a shell prompt.

Important

Do not edit the **`/etc/printcap`** file or the files in the **`/etc/cups/`** directory. Each time the printer daemon (**`cups`**) is started or restarted, new configuration files are dynamically created. The files are dynamically created when changes are applied with the **Printer Configuration Tool** as well.

The following types of print queues can be configured:

- **Locally-connected** — a printer attached directly to the computer through a parallel or USB port.
- **Networked CUPS (IPP)** — a printer that can be accessed over a TCP/IP network via the Internet Printing Protocol, also known as IPP (for example, a printer attached to another Red Hat Enterprise Linux system running CUPS on the network).
- **Networked UNIX (LPD)** — a printer attached to a different UNIX system that can be accessed over a TCP/IP network (for example, a printer attached to another Red Hat Enterprise Linux system running LPD on the network).
- **Networked Windows (SMB)** — a printer attached to a different system which is sharing a printer over a SMB network (for example, a printer attached to a Microsoft Windows™ machine).
- **Networked Novell (NCP)** — a printer attached to a different system which uses Novell's NetWare network technology.
- **Networked JetDirect** — a printer connected directly to the network through HP JetDirect instead of to a computer.

Important

If you add a new print queue or modify an existing one, you must apply the changes to them to take effect.

Clicking the **Apply** button saves any changes that you have made and restarts the printer daemon. The changes are not written to the configuration file until the printer daemon is restarted. Alternatively, you can choose **Action** => **Apply**.

For more information on the configuration of printers under Red Hat Enterprise Linux refer to the *System Administrators Guide*.

7.9. Additional Resources

Printing configuration and network printing are broad topics requiring knowledge and experience in hardware, networking, and system administration. For more detailed information about deploying printer services in your environments, refer to the following resources.

7.9.1. Installed Documentation

- **`lpr(1)`** man page -- Learn how to print selected files on the printer of your choice.
- **`lprm(1)`** man page -- Learn how to remove print jobs from a printer queue.
- **`cupsd(8)`** man page -- Learn about the CUPS printer daemon.
- **`cupsd.conf(5)`** man page -- Learn about the file format for the CUPS printer daemon configuration file.
- **`classes.conf(5)`** man page -- Learn about the file format for the CUPS class configuration file.
- Files in **`/usr/share/doc/cups-<version>`** -- Learn more about the CUPS printing system.

7.9.2. Useful Websites

- *http://www.webopedia.com/TERM/p/printer.html* -- General definitions of printers and descriptions of printer types.
- *http://www.linuxprinting.org/* -- A database of documents about printing, along with a database of nearly 1000 printers compatible with Linux printing facilities.
- *http://www.cups.org/* -- Documentation, FAQs, and newsgroups about CUPS.

7.9.3. Related Books

- *Network Printing* by Matthew Gast and Todd Radermacher; O'Reilly & Associates, Inc. -- Comprehensive information on using Linux as a print server in heterogeneous environments.
- The *System Administrators Guide*; Red Hat, Inc -- Includes a chapter on printer configuration.

Planning for Disaster

Disaster planning is a subject that is easy for a system administrator to forget -- it is not pleasant, and it always seems that there is something else more pressing to do. However, letting disaster planning slide is one of the worst things a system administrator can do.

Although it is often the dramatic disasters (such as a fire, flood, or storm) that first come to mind, the more mundane problems (such as construction workers cutting cables or even an overflowing sink) can be just as disruptive. Therefore, the definition of a disaster that a system administrator should keep in mind is any unplanned event that disrupts the normal operation of the organization.

While it would be impossible to list all the different types of disasters that could strike, this section examines the leading factors that are part of each type of disaster so that any possible exposure can be examined not in terms of its likelihood, but in terms of the factors that could lead to disaster.

8.1. Types of Disasters

In general, there are four different factors that can trigger a disaster. These factors are:

- Hardware failures
- Software failures
- Environmental failures
- Human errors

8.1.1. Hardware Failures

Hardware failures are easy to understand -- the hardware fails, and work grinds to a halt. What is more difficult to understand is the nature of the failures and how your exposure to them can be minimized. Here are some approaches that you can use:

8.1.1.1. Keeping Spare Hardware

At its simplest, exposure due to hardware failures can be reduced by having spare hardware available. Of course, this approach assumes two things:

- Someone on-site has the necessary skills to diagnose the problem, identify the failing hardware, and replace it.
- A replacement for the failing hardware is available.

These issues are covered in more detail in the following sections.

8.1.1.1.1. Having the Skills

Depending on your past experience and the hardware involved, having the necessary skills might be a non-issue. However, if you have not worked with hardware before, you might consider looking into local community colleges for introductory courses on PC repair. While such a course is not in and of itself sufficient to prepare you for tackling problems with an enterprise-level server, it is a good way to learn the basics (proper handling of tools and components, basic diagnostic procedures, and so on).

Note

Before taking the approach of first fixing it yourself, make sure that the hardware in question:

- Is not still under warranty
- Is not under a service/maintenance contract of any kind

If you attempt repairs on hardware that is covered by a warranty and/or service contract, you are likely violating the terms of these agreements and jeopardizing your continued coverage.

However, even with minimal skills, it might be possible to effectively diagnose and replace failing hardware -- if you choose your stock of replacement hardware properly.

8.1.1.1.2. What to Stock?

This question illustrates the multi-faceted nature of anything related to disaster recovery. When considering what hardware to stock, here are some of the issues you should keep in mind:

- Maximum allowable downtime
- The skill required to make the repair
- Budget available for spares
- Storage space required for spares
- Other hardware that could utilize the same spares

Each of these issues has a bearing on the types of spares that should be stocked. For example, stocking complete systems would tend to minimize downtime and require minimal skills to install but would be much more expensive than having a spare CPU and RAM module on a shelf. However, this expense might be worthwhile if your organization has several dozen identical servers that could benefit from a single spare system.

No matter what the final decision, the following question is inevitable and is discussed next.

8.1.1.1.2.1. How Much to Stock?

The question of spare stock levels is also multi-faceted. Here the main issues are:

- Maximum allowable downtime
- Projected rate of failure
- Estimated time to replenish stock
- Budget available for spares
- Storage space required for spares
- Other hardware that could utilize the same spares

At one extreme, for a system that can afford to be down a maximum of two days, and a spare that might be used once a year and could be replenished in a day, it would make sense to carry only one spare (and maybe even none, if you were confident of your ability to secure a spare within 24 hours).

At the other end of the spectrum, a system that can afford to be down no more than a few minutes, and a spare that might be used once a month (and could take several weeks to replenish) might mean that a half dozen spares (or more) should be on the shelf.

8.1.1.1.3. Spares That Are Not Spares

When is a spare not a spare? When it is hardware that is in day-to-day use but is also available to serve as a spare for a higher-priority system should the need arise. This approach has some benefits:

- Less money dedicated to "non-productive" spares
- The hardware is known to be operative

There are, however, downsides to this approach:

- Normal production of the lower-priority task is interrupted
- There is an exposure should the lower-priority hardware fail (leaving no spare for the higher-priority hardware)

Given these constraints, the use of another production system as a spare may work, but the success of this approach hinges on the system's specific workload and the impact the system's absence has on overall data center operations.

8.1.1.2. Service Contracts

Service contracts make the issue of hardware failures someone else's problem. All that is necessary for you to do is to confirm that a failure has, in fact, occurred and that it does not appear to have a software-related cause. You then make a telephone call, and someone shows up to make things right again.

It seems so simple. But as with most things in life, there is more to it than meets the eye. Here are some things that you must consider when looking at a service contract:

- Hours of coverage
- Response time
- Parts availability
- Available budget
- Hardware to be covered

We explore each of these details more closely in the following sections.

8.1.1.2.1. Hours of Coverage

Different service contracts are available to meet different needs; one of the big variables between different contracts relates to the hours of coverage. Unless you are willing to pay a premium for the privilege, you cannot just call any time and expect to see a technician at your door a short time later.

Instead, depending on your contract, you might find that you cannot even phone the service company until a specific day/time, or if you can, they will not dispatch a technician until the day/time specified for your contract.

Most hours of coverage are defined in terms of the hours and the days during which a technician may be dispatched. Some of the more common hours of coverage are:

- Monday through Friday, 09:00 to 17:00
- Monday through Friday, 12/18/24 hours each day (with the start and stop times mutually agreed upon)
- Monday through Saturday (or Monday through Sunday), same times as above

As you might expect, the cost of a contract increases with the hours of coverage. In general, extending the coverage Monday through Friday tends to cost less than adding on Saturday and Sunday coverage.

But even here there is a possibility of reducing costs if you are willing to do some of the work.

8.1.1.2.1.1. Depot Service

If your situation does not require anything more than the availability of a technician during standard business hours and you have sufficient experience to be able to determine what is broken, you might consider looking at *depot service*. Known by many names (including *walk-in service* and *drop-off service*), manufacturers may have service depots where technicians work on hardware brought in by customers.

Depot service has the benefit of being as fast as you are. You do not have to wait for a technician to become available and show up at your facility. Depot technicians do not go out on customer calls, meaning that there will be someone to work on your hardware as soon as you can get it to the depot.

Because depot service is done at a central location, there is a good chance that any required parts will be available. This can eliminate the need for an overnight shipment or waiting for a part to be driven several hundred miles from another office that just happened to have that part in stock.

There are some trade-offs, however. The most obvious is that you cannot choose the hours of service -- you get service when the depot is open. Another aspect to this is that the technicians do not work past their quitting time, so if your system failed at 16:30 on a Friday and you got the system to the depot by 17:00, it will not be worked on until the technicians arrive at work the following Monday morning.

Another trade-off is that depot service depends on having a depot nearby. If your organization is located in a metropolitan area, this is likely not going to be a problem. However, organizations in more rural locations may find that a depot is a long drive away.

> **Note**
>
> If considering depot service, take a moment and consider the mechanics of actually getting the hardware to the depot. Will you be using a company vehicle or your own? If your own, does your vehicle have the necessary space and load capacity? What about insurance? Will more than one person be necessary to load and unload the hardware?
>
> Although these are rather mundane concerns, they should be addressed before making the decision to use depot service.

8.1.1.2.2. Response Time

In addition to the hours of coverage, many service agreements specify a level of response time. In other words, when you call requesting service, how long will it be before a technician arrives? As you might imagine, a faster response time equates to a more expensive service agreement.

There are limits to the response times that are available. For instance, the travel time from the manufacturer's office to your facility has a large bearing on the response times that are possible[1]. Response times in the four hour range are usually considered among the quicker offerings. Slower response times can range from eight hours (which effectively becomes "next day" service for a standard business hours agreement), to 24 hours. As with every other aspect of a service agreement, even these times are negotiable -- for the right price.

> **Note**
>
> Although it is not a common occurrence, you should be aware that service agreements with response time clauses can sometimes stretch a manufacturer's service organization beyond its ability to respond. It is not unheard of for a very busy service organization to send somebody -- *anybody* -- on a short response-time service call just to meet their response time commitment. This person apparently diagnoses the problem, calling "the office" to have someone bring "the right part."
>
> In fact, they are just waiting until someone who is actually capable of handling the call arrives.
>
> While it might be understandable to see this happen under extraordinary circumstances (such as power problems that have damaged systems throughout their service area), if this is a consistent method of operation you should contact the service manager and demand an explanation.

If your response time needs are stringent (and your budget correspondingly large), there is one approach that can cut your response times even further -- to zero.

8.1.1.2.2.1. Zero Response Time -- Having an On-Site Technician

Given the appropriate situation (you are one of the biggest customers in the area), sufficient need (downtime of *any* magnitude is unacceptable), and financial resources (if you have to ask for the price, you probably cannot afford it), you might be a candidate for a full-time, on-site technician. The benefits of having a technician always standing by are obvious:

- Instant response to any problem
- A more proactive approach to system maintenance

As you might expect, this option can be *very* expensive, particularly if you require an on-site technician 24x7. But if this approach is appropriate for your organization, you should keep a number of points in mind in order to gain the most benefit.

First, on-site technicians need many of the resources of a regular employee, such as a workspace, telephone, appropriate access cards and/or keys, and so on.

On-site technicians are not very helpful if they do not have the proper parts. Therefore, make sure that secure storage is set aside for the technician's spare parts. In addition, make sure that the technician keeps a stock of parts appropriate for your configuration and that those parts are not routinely "cannibalized" by other technicians for their customers.

[1] And this would likely be considered a best-case response time, as technicians usually are responsible for territories that extend away from their office in all directions. If you are at one end of their territory and the only available technician is at the other end, the response time will be even longer.

8.1.1.2.3. Parts Availability

Obviously, the availability of parts plays a large role in limiting your organization's exposure to hardware failures. In the context of a service agreement, the availability of parts takes on another dimension, as the availability of parts applies not only to your organization, but to any other customer in the manufacturer's territory that might need those parts as well. Another organization that has purchased more of the manufacturer's hardware than you might get preferential treatment when it comes to getting parts (and technicians, for that matter).

Unfortunately, there is little that can be done in such circumstances, short of working out the problem with the service manager.

8.1.1.2.4. Available Budget

As outlined above, service contracts vary in price according to the nature of the services being provided. Keep in mind that the costs associated with a service contract are a recurring expense; each time the contract is due to expire you must negotiate a new contract and pay again.

8.1.1.2.5. Hardware to be Covered

Here is an area where you might be able to help keep costs to a minimum. Consider for a moment that you have negotiated a service agreement that has an on-site technician 24x7, on-site spares -- you name it. Every single piece of hardware you have purchased from this vendor is covered, including the PC that the company receptionist uses for non-critical tasks.

Does that PC *really* need to have someone on-site 24x7? Even if the PC is vital to the receptionist's job, the receptionist only works from 09:00 to 17:00; it is highly unlikely that:

- The PC will be in use from 17:00 to 09:00 the next morning (not to mention weekends)
- A failure of this PC will be noticed, except between 09:00 and 17:00

Therefore, paying on the chance that this PC might need to be serviced in the middle of a Saturday night is a waste of money.

The thing to do is to split up the service agreement such that non-critical hardware is grouped separately from more critical hardware. In this way, costs can be kept as low as possible.

Note

If you have twenty identically-configured servers that are critical to your organization, you might be tempted to have a high-level service agreement written for only one or two, with the rest covered by a much less expensive agreement. Then, the reasoning goes, no matter which one of the servers fails on a weekend, you will say that *it* is the one eligible for high-level service.

Do not do this. Not only is it dishonest, most manufacturers keep track of such things by using serial numbers. Even if you figure out a way around such checks, far more is spent after being discovered than by being honest and paying for the service you really need.

8.1.2. Software Failures

Software failures can result in extended downtimes. For example, owners of a certain brand of computer systems noted for their high-availability features recently experienced this firsthand. A bug in the time handling code of the computer's operating system resulted in each customer's systems crashing at a certain time of a certain day. While this particular situation is a more spectacular example of a software failure in action, other software-related failures may be less dramatic, but still as devastating.

Software failures can strike in one of two areas:

- Operating system
- Applications

Each type of failure has its own specific impact and is explored in more detail in the following sections.

8.1.2.1. Operating System Failures

In this type of failure, the operating system is responsible for the disruption in service. Operating system failures come from two areas:

- Crashes
- Hangs

The main thing to keep in mind about operating system failures is that they take out everything that the computer was running at the time of the failure. As such, operating system failures can be devastating to production.

8.1.2.1.1. Crashes

Crashes occur when the operating system experiences an error condition from which it cannot recover. The reasons for crashes can range from an inability to handle an underlying hardware problem to a bug in the kernel-level code comprising the operating system. When an operating system crashes, the system must be rebooted in order to continue production.

8.1.2.1.2. Hangs

When the operating system stops handling system events, the system grinds to a halt. This is known as a *hang*. Hangs can be caused by *deadlocks* (two resource consumers contending for resources the other has) and *livelocks* (two or more processes responding to each other's activities, but doing no useful work), but the end result is the same -- a complete lack of productivity.

8.1.2.2. Application Failures

Unlike operating system failures, application failures can be more limited in the scope of their damage. Depending on the specific application, a single application failing might impact only one person. On the other hand, if it is a server application servicing a large population of client applications, the consequences of a failure would be much more widespread.

Application failures, like operating system failures, can be due to hangs and crashes; the only difference is that here it is the application that is hanging or crashing.

8.1.2.3. Getting Help -- Software Support

Just as hardware vendors provide support for their products, many software vendors make support packages available to their customers. Except for the obvious differences (no spare hardware is required, and most of the work can be done by support personnel over the phone), software support contracts can be quite similar to hardware support contracts.

The level of support provided by a software vendor can vary. Here are some of the more common support strategies employed today:

- Documentation
- Self support
- Web or email support
- Telephone support
- On-site support

Each type of support is described in more detail in the following sections.

8.1.2.3.1. Documentation

Although often overlooked, software documentation can serve as a first-level support tool. Whether online or printed, documentation often contains the information necessary to resolve many issues.

8.1.2.3.2. Self Support

Self support relies on the customer using online resources to resolve their own software-related issues. Quite often these resources take the form of Web-based FAQs (Frequently Asked Questions) or knowledge bases.

FAQs often have little or no selection capabilities, leaving the customer to scroll through question after question in the hopes of finding one that addresses the issue at hand. Knowledge bases tend to be somewhat more sophisticated, allowing the entry of search terms. Knowledge bases can also be quite extensive in scope, making it a good tool for resolving problems.

8.1.2.3.3. Web or Email Support

Many times what looks like a self support website also includes Web-based forms or email addresses that make it possible to send questions to support staff. While this might at first glance appear to be an improvement over a good self support website, it really depends on the people answering the email.

If the support staff is overworked, it is difficult to get the necessary information from them, as their main concern is to quickly respond to each email and move on to the next one. The reason for this is because nearly all support personnel are evaluated by the number of issues that they resolve. Escalation of issues is also difficult because there is little that can be done within an email to encourage more timely and helpful responses -- particularly when the person reading your email is in a hurry to move on to the next one.

The way to get the best service is to make sure that your email addresses all the questions that a support technician might ask, such as:

- Clearly describe the nature of the problem
- Include all pertinent version numbers
- Describe what you have already done in an attempt to address the problem (applied the latest patches, rebooted with a minimal configuration, etc.)

By giving the support technician more information, you stand a better chance of getting the support you need.

8.1.2.3.4. Telephone Support

As the name implies, telephone support entails speaking to a support technician via telephone. This style of support is most similar to hardware support; that there can be various levels of support available (with different hours of coverage, response time, etc.).

8.1.2.3.5. On-Site Support

Also known as on-site consulting, on-site software support is normally reserved for resolving specific issues or making critical changes, such as initial software installation and configuration, major upgrades, and so on. As expected, this is the most expensive type of software support available.

Still, there are instances where on-site support makes sense. As an example, consider a small organization with a single system administrator. The organization is going to be deploying its first database server, but the deployment (and the organization) is not large enough to justify hiring a dedicated database administrator. In this situation, it can often be cheaper to bring in a specialist from the database vendor to handle the initial deployment (and occasionally later on, as the need arises) than it would be to train the system administrator in a skill that will be seldom used.

8.1.3. Environmental Failures

Even though the hardware may be running perfectly, and even though the software may be configured properly and is working as it should, problems can still occur. The most common problems that occur outside of the system itself have to do with the physical environment in which the system resides.

Environmental issues can be broken into four major categories:

- Building integrity
- Electricity
- Air conditioning
- Weather and the outside world

8.1.3.1. Building Integrity

For such a seemingly simple structure, a building performs a great many functions. It provides shelter from the elements. It provides the proper micro-climate for the building's contents. It has mechanisms to provide power and to protect against fire, theft, and vandalism. Performing all these functions, it is not surprising that there is a great deal that can go wrong with a building. Here are some possibilities to consider:

- Roofs can leak, allowing water into data centers.
- Various building systems (such as water, sewer, or air handling) can fail, rendering the building uninhabitable.
- Floors may have insufficient load-bearing capacity to hold the equipment you want to put in the data center.

It is important to have a creative mind when it comes to thinking about the different ways buildings can fail. The list above is only meant to start you thinking along the proper lines.

8.1.3.2. Electricity

Because electricity is the lifeblood of any computer system, power-related issues are paramount in the mind of system administrators everywhere. There are several different aspects to power; they are covered in more detail in the following sections.

8.1.3.2.1. The Security of Your Power

First, it is necessary to determine how secure your normal power supply may be. Just like nearly every other data center, you probably obtain your power from a local power company via power transmission lines. Because of this, there are limits to what you can do to make sure that your primary power supply is as secure as possible.

> **Note**
>
> Organizations located near the boundaries of a power company might be able to negotiate connections to two different power grids:
>
> - The one servicing your area
> - The one from the neighboring power company
>
> The costs involved in running power lines from the neighboring grid are sizable, making this an option only for larger organizations. However, such organizations find that the redundancy gained outweigh the costs in many cases.

The main things to check are the methods by which the power is brought onto your organization's property and into the building. Are the transmission lines above ground or below? Above-ground lines are susceptible to:

- Damage from extreme weather conditions (ice, wind, lightning)
- Traffic accidents that damage the poles and/or transformers
- Animals straying into the wrong place and shorting out the lines

However, below-ground lines have their own unique shortcomings:

- Damage from construction workers digging in the wrong place
- Flooding
- Lightning (though much less so than above-ground lines)

Continue to trace the power lines into your building. Do they first go to an outside transformer? Is that transformer protected from vehicles backing into it or trees falling on it? Are all exposed shutoff switches protected against unauthorized use?

Once inside your building, could the power lines (or the panels to which they attach) be subject to other problems? For instance, could a plumbing problem flood the electrical room?

Continue tracing the power into the data center; is there anything else that could unexpectedly interrupt your power supply? For example, is the data center sharing one or more circuits with non-data center loads? If so, the external load might one day trip the circuit's overload protection, taking down the data center as well.

8.1.3.2.2. Power Quality

It is not enough to ensure that the data center's power source is as secure as possible. You must also be concerned with the quality of the power being distributed throughout the data center. There are several factors that must be considered:

Voltage

The voltage of the incoming power must be stable, with no voltage reductions (often called *sags*, *droops*, or *brownouts*) or voltage increases (often known as *spikes* and *surges*).

Waveform

The waveform must be a clean sine wave, with minimal *THD* (Total Harmonic Distortion).

Frequency

The frequency must be stable (most countries use a power frequency of either 50Hz or 60Hz).

Noise

The power must not include any *RFI* (Radio Frequency Interference) or *EMI* (Electro-Magnetic Interference) noise.

Current

The power must be supplied at a current rating sufficient to run the data center.

Power supplied directly from the power company does not normally meet the standards necessary for a data center. Therefore, some level of power conditioning is usually required. There are several different approaches possible:

Surge Protectors

Surge protectors do just what their name implies -- they filter surges from the power supply. Most do nothing else, leaving equipment vulnerable to damage from other power-related problems.

Power Conditioners

Power conditioners attempt a more comprehensive approach; depending on the sophistication of the unit, power conditioners often can take care of most of the types of problems outlined above.

Motor-Generator Sets

A motor-generator set is essentially a large electric motor powered by your normal power supply. The motor is attached to a large flywheel, which is, in turn, attached to a generator. The motor turns the flywheel and generator, which generates electricity in sufficient quantities to run the data center. In this way, the data center power is electrically isolated from outside power, meaning that most power-related problems are eliminated. The flywheel also provides the ability to maintain power through short outages, as it takes several seconds for the flywheel to slow to the point at which it can no longer generate power.

Uninterruptible Power Supplies

Some types of Uninterruptible Power Supplies (more commonly known as a *UPS*) include most (if not all) of the protection features of a power conditioner[2].

With the last two technologies listed above, we have started in on the topic most people think of when they think about power -- backup power. In the next section, different approaches to providing backup power are explored.

[2] UPS technology is discussed in more detail in *Section 8.1.3.2.3.2, “Providing Power For the Next Few Minutes”*.

8.1.3.2.3. Backup Power

One power-related term that nearly everyone has heard is the term *blackout*. A blackout is a complete loss of electrical power and may last from a fraction of a second to weeks.

Because the length of blackouts can vary so greatly, it is necessary to approach the task of providing backup power using different technologies for power outages of different lengths.

> **Note**
>
> The most frequent blackouts last, on average, no more than a few seconds; longer outages are much less frequent. Therefore, concentrate first on protecting against blackouts of only a few minutes in duration, then work out methods of reducing your exposure to longer outages.

8.1.3.2.3.1. Providing Power For the Next Few Seconds

Since the majority of outages last only a few seconds, your backup power solution must have two primary characteristics:

- Very short time to switch to backup power (known as *transfer time*)
- A *runtime* (the time that backup power will last) measured in seconds to minutes

The backup power solutions that match these characteristics are motor-generator sets and UPSs. The flywheel in the motor-generator set allows the generator to continue producing electricity for enough time to ride out outages of a second or so. Motor-generator sets tend to be quite large and expensive, making them a practical solution only for mid-sized and larger data centers.

However, another technology -- called a UPS -- can fill in for those situations where a motor-generator set is too expensive. It can also handle longer outages.

8.1.3.2.3.2. Providing Power For the Next Few Minutes

UPSs can be purchased in a variety of sizes -- small enough to run a single low-end PC for five minutes or large enough to power an entire data center for an hour or more.

UPSs are made up of the following parts:

- A *transfer switch* for switching from the primary power supply to the backup power supply
- A battery, for providing backup power
- An *inverter*, which converts the DC current from the battery into the AC current required by the data center hardware

Apart from the size and battery capacity of the unit, UPSs come in two basic types:

- The *offline* UPS uses its inverter to generate power only when the primary power supply fails.
- The *online* UPS uses its inverter to generate power all the time, powering the inverter via its battery only when the primary power supply fails.

Each type has their advantages and disadvantages. The offline UPS is usually less expensive, because the inverter does not have to be constructed for full-time operation. However, a problem in the inverter of an offline UPS will go unnoticed (until the next power outage, that is).

Online UPSs tend to be better at providing clean power to your data center; after all, an online UPS is essentially generating power for you full time.

But no matter what type of UPS you choose, you must properly size the UPS to your anticipated load (thereby ensuring that the UPS has sufficient capacity to produce electricity at the required voltage and current), *and* you must determine how long you would like to be able to run your data center on battery power.

To determine this information, you must first identify those loads that are to be serviced by the UPS. Go to each piece of equipment and determine how much power it draws (this is normally listed on a label near the unit's power cord). Write down the voltage, watts, and/or amps. Once you have these figures for all of the hardware, you must convert them to *VA* (Volt-Amps). If you have a wattage number, you can use the listed wattage as the VA; if you have amps, multiply it by volts to get VA. By adding the VA figures you can arrive at the approximate VA rating required for the UPS.

Note

Strictly speaking, this approach to calculating VA is not entirely correct; however, to get the true VA you would need to know the power factor for each unit, and this information is rarely, if ever, provided. In any case, the VA numbers obtained from this approach reflects worst-case values, leaving a large margin of error for safety.

Determining runtime is more of a business question than a technical question -- what sorts of outages are you willing to protect against, and how much money are you prepared to spend to do so? Most sites select runtimes that are less than an hour or two at most, as battery-backed power becomes very expensive beyond this point.

8.1.3.2.3.3. Providing Power For the Next Few Hours (and Beyond)

Once we get into power outages that are measured in days, the choices get even more expensive. The technologies capable of handling long-term power outages are limited to generators powered by some type of engine -- diesel and gas turbine, primarily.

Note

Keep in mind that engine-powered generators require regular refueling while they are running. You should know your generator's fuel "burn" rate at maximum load and arrange fuel deliveries accordingly.

At this point, your options are wide open, assuming your organization has sufficient funds. This is also an area where experts should help you determine the best solution for your organization. Very few system administrators have the specialized knowledge necessary to plan the acquisition and deployment of these kinds of power generation systems.

Note

Portable generators of all sizes can be rented, making it possible to have the benefits of generator power without the initial outlay of money necessary to purchase one. However, keep in mind that in disasters affecting your general vicinity, rented generators will be in very short supply and very expensive.

8.1.3.2.4. Planning for Extended Outages

While a black out of five minutes is little more than an inconvenience to the personnel in a darkened office, what about an outage that lasts an hour? Five hours? A day? A week?

The fact is, even if the data center is operating normally, an extended outage will eventually affect your organization at some point. Consider the following points:

- What if there is no power to maintain environmental control in the data center?
- What if there is no power to maintain environmental control in the entire building?
- What if there is no power to operate personal workstations, the telephone system, the lights?

The point here is that your organization must determine at what point an extended outage will just have to be tolerated. Or if that is not an option, your organization must reconsider its ability to function completely independently of on-site power for extended periods, meaning that very large generators will be needed to power the entire building.

Of course, even this level of planning cannot take place in a vacuum. It is very likely that whatever caused the extended outage is also affecting the world outside your organization, and that the outside world will start having an affect on your organization's ability to continue operations, even given unlimited power generation capacity.

8.1.3.3. Heating, Ventilation, and Air Conditioning

The Heating, Ventilation, and Air Conditioning (*HVAC*) systems used in today's office buildings are incredibly sophisticated. Often computer controlled, the HVAC system is vital to providing a comfortable work environment.

Data centers usually have additional air handling equipment, primarily to remove the heat generated by the many computers and associated equipment. Failures in an HVAC system can be devastating to the continued operation of a data center. And given their complexity and electro-mechanical nature, the possibilities for failure are many and varied. Here are a few examples:

- The air handling units (essentially large fans driven by large electric motors) can fail due to electrical overload, bearing failure, belt/pulley failure, etc.
- The cooling units (often called *chillers*) can lose their refrigerant due to leaks, or they can have their compressors and/or motors seize.

HVAC repair and maintenance is a very specialized field -- a field that the average system administrator should leave to the experts. If anything, a system administrator should make sure that the HVAC equipment serving the data center is checked for normal operation on a daily basis (if not more frequently) and is maintained according to the manufacturer's guidelines.

8.1.3.4. Weather and the Outside World

There are some types of weather that can cause problems for a system administrator:

- Heavy snow and ice can prevent personnel from getting to the data center, and can even clog air conditioning condensers, resulting in elevated data center temperatures just when no one is able to get to the data center to take corrective action.
- High winds can disrupt power and communications, with extremely high winds actually doing damage to the building itself.

There are other types of weather than can still cause problems, even if they are not as well known. For example, exceedingly high temperatures can result in overburdened cooling systems, and brownouts or blackouts as the local power grid becomes overloaded.

Although there is little that can be done about the weather, knowing the way that it can affect your data center operations can help you to keep things running even when the weather turns bad.

8.1.4. Human Errors

It has been said that computers really *are* perfect. The reasoning behind this statement is that if you dig deeply enough, behind every computer error you will find the human error that caused it. In this section, the more common types of human errors and their impacts are explored.

8.1.4.1. End-User Errors

The users of a computer can make mistakes that can have serious impact. However, due to their normally unprivileged operating environment, user errors tend to be localized in nature. Because most users interact with a computer exclusively through one or more applications, it is within applications that most end-user errors occur.

8.1.4.1.1. Improper Use of Applications

When applications are used improperly, various problems can occur:

- Files inadvertently overwritten
- Wrong data used as input to an application
- Files not clearly named and organized
- Files accidentally deleted

The list could go on, but this is enough to illustrate the point. Due to users not having super-user privileges, the mistakes they make are usually limited to their own files. As such, the best approach is two-pronged:

- Educate users in the proper use of their applications and in proper file management techniques
- Make sure backups of users' files are made regularly and that the restoration process is as streamlined and quick as possible

Beyond this, there is little that can be done to keep user errors to a minimum.

8.1.4.2. Operations Personnel Errors

Operators have a more in-depth relationship with an organization's computers than end-users. Where end-user errors tend to be application-oriented, operators tend to perform a wider range of tasks. Although the nature of the tasks have been dictated by others, some of these tasks can include the use of system-level utilities, where the potential for widespread damage due to errors is greater. Therefore, the types of errors that an operator might make center on the operator's ability to follow the procedures that have been developed for the operator's use.

8.1.4.2.1. Failure to Follow Procedures

Operators should have sets of procedures documented and available for nearly every action they perform[3]. It might be that an operator does not follow the procedures as they are laid out. There can be several reasons for this:

- The environment was changed at some time in the past, and the procedures were never updated. Now the environment changes again, rendering the operator's memorized procedure invalid. At this point, even if the procedures were updated (which is unlikely, given the fact that they were not updated before) the operator will not be aware of it.

- The environment was changed, and no procedures exist. This is just a more out-of-control version of the previous situation.

- The procedures exist and are correct, but the operator will not (or cannot) follow them.

Depending on the management structure of your organization, you might not be able to do much more than communicate your concerns to the appropriate manager. In any case, making yourself available to do what you can to help resolve the problem is the best approach.

8.1.4.2.2. Mistakes Made During Procedures

Even if the operator follows the procedures, and even if the procedures are correct, it is still possible for mistakes to be made. If this happens, the possibility exists that the operator is careless (in which case the operator's management should become involved).

Another explanation is that it was just a mistake. In these cases, the best operators realize that something is wrong and seek assistance. Always encourage the operators you work with to contact the appropriate people immediately if they suspect something is wrong. Although many operators are highly-skilled and able to resolve many problems independently, the fact of the matter is that this is not their job. And a problem that is made worse by a well-meaning operator harms both that person's career and your ability to quickly resolve what might originally have been a small problem.

8.1.4.3. System Administrator Errors

Unlike operators, system administrators perform a wide variety of tasks using an organization's computers. Also unlike operators, the tasks that system administrators perform are often not based on documented procedures.

Therefore, system administrators sometimes make unnecessary work for themselves when they are not careful about what they are doing. During the course of carrying out day-to-day responsibilities, system administrators have more than sufficient access to the computer systems (not to mention their super-user access privileges) to mistakenly bring systems down.

System administrators either make errors of misconfiguration or errors during maintenance.

8.1.4.3.1. Misconfiguration Errors

System administrators must often configure various aspects of a computer system. This configuration might include:

- Email

[3] If the operators at your organization do not have a set of operating procedures, work with them, your management, and your users to get them created. Without them, a data center is out of control and likely to experience severe problems in the course of day-to-day operations.

- User accounts
- Network
- Applications

The list could go on quite a bit longer. The actual task of configuration varies greatly; some tasks require editing a text file (using any one of a hundred different configuration file syntaxes), while other tasks require running a configuration utility.

The fact that these tasks are all handled differently is merely an additional challenge to the basic fact that each configuration task itself requires different knowledge. For example, the knowledge required to configure a mail transport agent is fundamentally different from the knowledge required to configure a new network connection.

Given all this, perhaps it should be surprising that so *few* mistakes are actually made. In any case, configuration is, and will continue to be, a challenge for system administrators. Is there anything that can be done to make the process less error-prone?

8.1.4.3.1.1. Change Control

The common thread of every configuration change is that some sort of a change is being made. The change may be large, or it may be small. But it is still a change and should be treated in a particular way.

Many organizations implement some type of change control process. The intent is to help system administrators (and all parties affected by the change) to manage the process of change and to reduce the organization's exposure to any errors that may occur.

A change control process normally breaks the change into different steps. Here is an example:

Preliminary research

> Preliminary research attempts to clearly define:
>
> - The nature of the change to take place
> - Its impact, should the change succeed
> - A fallback position, should the change fail
> - An assessment of what types of failures are possible
>
> Preliminary research might include testing the proposed change during a scheduled downtime, or it may go so far as to include implementing the change first on a special test environment run on dedicated test hardware.

Scheduling

> The change is examined with an eye toward the actual mechanics of implementation. The scheduling being done includes outlining the sequencing and timing of the change (along with the sequencing and timing of any steps necessary to back the change out should a problem arise), as well as ensuring that the time allotted for the change is sufficient and does not conflict with any other system-level activity.
>
> The product of this process is often a checklist of steps for the system administrator to use while making the change. Included with each step are instructions to perform in order to back out the change should the step fail. Estimated times are often included, making it easier for the system administrator to determine whether the work is on schedule or not.

Execution

At this point, the actual execution of the steps necessary to implement the change should be straightforward and anti-climactic. The change is either implemented, or (if trouble crops up) it is backed out.

Monitoring

Whether the change is implemented or not, the environment is monitored to make sure that everything is operating as it should.

Documenting

If the change has been implemented, all existing documentation is updated to reflect the changed configuration.

Obviously, not all configuration changes require this level of detail. Creating a new user account should not require any preliminary research, and scheduling would likely consist of determining whether the system administrator has a spare moment to create the account. Execution would be similarly quick; monitoring might consist of ensuring that the account was usable, and documenting would probably entail sending an email to the new user's manager.

But as configuration changes become more complex, a more formal change control process becomes necessary.

8.1.4.3.2. Mistakes Made During Maintenance

This type of error can be insidious because there is usually so little planning and tracking done during day-to-day maintenance.

System administrators see the results of this kind of error every day, especially from the many users that swear they did not change a thing -- the computer just broke. The user that says this usually does not remember what they did, and when the same thing happens to you, you may not remember what you did, either.

The key thing to keep in mind is that you must be able to remember what changes you made during maintenance if you are to be able to resolve any problems quickly. A full-blown change control process is not realistic for the hundreds of small things done over the course of a day. What can be done to keep track of the 101 small things a system administrator does every day?

The answer is simple -- takes notes. Whether it is done in a paper notebook, a PDA, or as comments in the affected files, take notes. By tracking what you have done, you stand a better chance of seeing a failure as being related to a change you recently made.

8.1.4.4. Service Technician Errors

Sometimes the very people that are supposed to help you keep your systems running reliably can actually make things worse. This is not due to any conspiracy; it is just that anyone working on any technology for any reason risks rendering that technology inoperable. The same effect is at work when programmers fix one bug but end up creating another.

8.1.4.4.1. Improperly-Repaired Hardware

In this case, the technician either failed to correctly diagnose the problem and made an unnecessary (and useless) repair, or the diagnosis was correct, but the repair was not carried out properly. It may be that the replacement part was itself defective, or that the proper procedure was not followed when the repair was carried out.

This is why it is important to be aware of what the technician is doing at all times. By doing this, you can keep an eye out for failures that seem to be related to the original problem in some way.

This keeps the technician on track should there be a problem; otherwise there is a chance that the technician will view this fault as being new and unrelated to the one that was supposedly fixed. In this way, time is not wasted chasing the wrong problem.

8.1.4.4.2. Fixing One Thing and Breaking Another

Sometimes, even though a problem was diagnosed and repaired successfully, another problem pops up to take its place. The CPU module was replaced, but the anti-static bag it came in was left in the cabinet, blocking the fan and causing an over-temperature shutdown. Or the failing disk drive in the RAID array was replaced, but because a connector on another drive was bumped and accidentally disconnected, the array is still down.

These things might be the result of chronic carelessness or an honest mistake. It does not matter. What you should always do is to carefully review the repairs made by the technician and ensure that the system is working properly before letting the technician leave.

8.2. Backups

Backups have two major purposes:

- To permit restoration of individual files
- To permit wholesale restoration of entire file systems

The first purpose is the basis for the typical file restoration request: a user accidentally deletes a file and asks that it be restored from the latest backup. The exact circumstances may vary somewhat, but this is the most common day-to-day use for backups.

The second situation is a system administrator's worst nightmare: for whatever reason, the system administrator is staring at hardware that used to be a productive part of the data center. Now, it is little more than a lifeless chunk of steel and silicon. The thing that is missing is all the software and data you and your users have assembled over the years. Supposedly everything has been backed up. The question is: has it?

And if it has, can you restore it?

8.2.1. Different Data: Different Backup Needs

Look at the kinds of data[4] processed and stored by a typical computer system. Notice that some of the data hardly ever changes, and some of the data is constantly changing.

The pace at which data changes is crucial to the design of a backup procedure. There are two reasons for this:

- A backup is nothing more than a snapshot of the data being backed up. It is a reflection of that data at a particular moment in time.
- Data that changes infrequently can be backed up infrequently, while data that changes often must be backed up more frequently.

[4] We are using the term *data* in this section to describe anything that is processed via backup software. This includes operating system software, application software, as well as actual data. No matter what it is, as far as backup software is concerned, it is all data.

System administrators that have a good understanding of their systems, users, and applications should be able to quickly group the data on their systems into different categories. However, here are some examples to get you started:

Operating System

> This data normally only changes during upgrades, the installation of bug fixes, and any site-specific modifications.
>
> **Note**
>
> Should you even bother with operating system backups? This is a question that many system administrators have pondered over the years. On the one hand, if the installation process is relatively easy, and if the application of bugfixes and customizations are well documented and easily reproducible, reinstalling the operating system may be a viable option.
>
> On the other hand, if there is the least doubt that a fresh installation can completely recreate the original system environment, backing up the operating system is the best choice, even if the backups are performed much less frequently than the backups for production data. Occasional operating system backups also come in handy when only a few system files must be restored (for example, due to accidental file deletion).

Application Software

> This data changes whenever applications are installed, upgraded, or removed.

Application Data

> This data changes as frequently as the associated applications are run. Depending on the specific application and your organization, this could mean that changes take place second-by-second or once at the end of each fiscal year.

User Data

> This data changes according to the usage patterns of your user community. In most organizations, this means that changes take place all the time.

Based on these categories (and any additional ones that are specific to your organization), you should have a pretty good idea concerning the nature of the backups that are needed to protect your data.

Note

You should keep in mind that most backup software deals with data on a directory or file system level. In other words, your system's directory structure plays a part in how backups will be performed. This is another reason why it is always a good idea to carefully consider the best directory structure for a new system and group files and directories according to their anticipated usage.

8.2.2. Backup Software: Buy Versus Build

In order to perform backups, it is first necessary to have the proper software. This software must not only be able to perform the basic task of making copies of bits onto backup media, it must also interface cleanly with your organization's personnel and business needs. Some of the features to consider when reviewing backup software include:

- Schedules backups to run at the proper time

- Manages the location, rotation, and usage of backup media
- Works with operators (and/or robotic media changers) to ensure that the proper media is available
- Assists operators in locating the media containing a specific backup of a given file

As you can see, a real-world backup solution entails much more than just scribbling bits onto your backup media.

Most system administrators at this point look at one of two solutions:

- Purchase a commercially-developed solution
- Create an in-house developed backup system from scratch (possibly integrating one or more open source technologies)

Each approach has its good and bad points. Given the complexity of the task, an in-house solution is not likely to handle some aspects (such as media management, or have comprehensive documentation and technical support) very well. However, for some organizations, this might not be a shortcoming.

A commercially-developed solution is more likely to be highly functional, but may also be overly-complex for the organization's present needs. That said, the complexity might make it possible to stick with one solution even as the organization grows.

As you can see, there is no clear-cut method for deciding on a backup system. The only guidance that can be offered is to ask you to consider these points:

- Changing backup software is difficult; once implemented, you will be using the backup software for a long time. After all, you will have long-term archive backups that you must be able to read. Changing backup software means you must either keep the original software around (to access the archive backups), or you must convert your archive backups to be compatible with the new software.

  Depending on the backup software, the effort involved in converting archive backups may be as straightforward (though time-consuming) as running the backups through an already-existing conversion program, or it may require reverse-engineering the backup format and writing custom software to perform the task.
- The software must be 100% reliable -- it must back up what it is supposed to, when it is supposed to.
- When the time comes to restore any data -- whether a single file or an entire file system -- the backup software must be 100% reliable.

8.2.3. Types of Backups

If you were to ask a person that was not familiar with computer backups, most would think that a backup was just an identical copy of *all* the data on the computer. In other words, if a backup was created Tuesday evening, and nothing changed on the computer all day Wednesday, the backup created Wednesday evening would be identical to the one created on Tuesday.

While it is possible to configure backups in this way, it is likely that you would not. To understand more about this, we must first understand the different types of backups that can be created. They are:

- Full backups
- Incremental backups

- Differential backups

8.2.3.1. Full Backups

The type of backup that was discussed at the beginning of this section is known as a *full backup*. A full backup is a backup where every single file is written to the backup media. As noted above, if the data being backed up never changes, every full backup being created will be the same.

That similarity is due to the fact that a full backup does not check to see if a file has changed since the last backup; it blindly writes everything to the backup media whether it has been modified or not.

This is the reason why full backups are not done all the time -- every file is written to the backup media. This means that a great deal of backup media is used even if nothing has changed. Backing up 100 gigabytes of data each night when maybe 10 megabytes worth of data has changed is not a sound approach; that is why *incremental backups* were created.

8.2.3.2. Incremental Backups

Unlike full backups, incremental backups first look to see whether a file's modification time is more recent than its last backup time. If it is not, the file has not been modified since the last backup and can be skipped this time. On the other hand, if the modification date *is* more recent than the last backup date, the file has been modified and should be backed up.

Incremental backups are used in conjunction with a regularly-occurring full backup (for example, a weekly full backup, with daily incrementals).

The primary advantage gained by using incremental backups is that the incremental backups run more quickly than full backups. The primary disadvantage to incremental backups is that restoring any given file may mean going through one or more incremental backups until the file is found. When restoring a complete file system, it is necessary to restore the last full backup and every subsequent incremental backup.

In an attempt to alleviate the need to go through every incremental backup, a slightly different approach was implemented. This is known as the *differential backup*.

8.2.3.3. Differential Backups

Differential backups are similar to incremental backups in that both backup only modified files. However, differential backups are *cumulative* -- in other words, with a differential backup, once a file has been modified it continues to be included in all subsequent differential backups (until the next, full backup, of course).

This means that each differential backup contains all the files modified since the last full backup, making it possible to perform a complete restoration with only the last full backup and the last differential backup.

Like the backup strategy used with incremental backups, differential backups normally follow the same approach: a single periodic full backup followed by more frequent differential backups.

The effect of using differential backups in this way is that the differential backups tend to grow a bit over time (assuming different files are modified over the time between full backups). This places differential backups somewhere between incremental backups and full backups in terms of backup media utilization and backup speed, while often providing faster single-file and complete restorations (due to fewer backups to search/restore).

Given these characteristics, differential backups are worth careful consideration.

8.2.4. Backup Media

We have been very careful to use the term "backup media" throughout the previous sections. There is a reason for that. Most experienced system administrators usually think about backups in terms of reading and writing tapes, but today there are other options.

At one time, tape devices were the only removable media devices that could reasonably be used for backup purposes. However, this has changed. In the following sections we look at the most popular backup media, and review their advantages as well as their disadvantages.

8.2.4.1. Tape

Tape was the first widely-used removable data storage medium. It has the benefits of low media cost and reasonably-good storage capacity. However, tape has some disadvantages -- it is subject to wear, and data access on tape is sequential in nature.

These factors mean that it is necessary to keep track of tape usage (retiring tapes once they have reached the end of their useful life), and that searching for a specific file on tape can be a lengthy proposition.

On the other hand, tape is one of the most inexpensive mass storage media available, and it has a long history of reliability. This means that building a good-sized tape library need not consume a large part of your budget, and you can count on it being usable now and in the future.

8.2.4.2. Disk

In years past, disk drives would never have been used as a backup medium. However, storage prices have dropped to the point where, in some cases, using disk drives for backup storage does make sense.

The primary reason for using disk drives as a backup medium would be speed. There is no faster mass storage medium available. Speed can be a critical factor when your data center's backup window is short, and the amount of data to be backed up is large.

But disk storage is not the ideal backup medium, for a number of reasons:

- Disk drives are not normally removable. One key factor to an effective backup strategy is to get the backups out of your data center and into off-site storage of some sort. A backup of your production database sitting on a disk drive two feet away from the database itself is not a backup; it is a copy. And copies are not very useful should the data center and its contents (including your copies) be damaged or destroyed by some unfortunate set of circumstances.

- Disk drives are expensive (at least compared to other backup media). There may be situations where money truly is no object, but in all other circumstances, the expenses associated with using disk drives for backup mean that the number of backup copies must be kept low to keep the overall cost of backups low. Fewer backup copies mean less redundancy should a backup not be readable for some reason.

- Disk drives are fragile. Even if you spend the extra money for removable disk drives, their fragility can be a problem. If you drop a disk drive, you have lost your backup. It is possible to purchase specialized cases that can reduce (but not entirely eliminate) this hazard, but that makes an already-expensive proposition even more so.

- Disk drives are not archival media. Even assuming you are able to overcome all the other problems associated with performing backups onto disk drives, you should consider the following. Most organizations have various legal requirements for keeping records available for certain lengths of time. The chance of getting usable data from a 20-year-old tape is much greater than the chance of getting usable data from a 20-year-old disk drive. For instance, would you still have the hardware necessary to connect it to your system? Another thing to consider is that a disk drive is much more complex than a tape cartridge. When a 20-year-old motor spins a 20-year-old disk platter, causing 20-year-old read/write heads to fly over the platter surface, what are the chances that all these components will work flawlessly after sitting idle for 20 years?

> **Note**
>
> Some data centers back up to disk drives and then, when the backups have been completed, the backups are written out to tape for archival purposes. This allows for the fastest possible backups during the backup window. Writing the backups to tape can then take place during the remainder of the business day; as long as the "taping" finishes before the next day's backups are done, time is not an issue.

All this said, there are still some instances where backing up to disk drives might make sense. In the next section we see how they can be combined with a network to form a viable (if expensive) backup solution.

8.2.4.3. Network

By itself, a network cannot act as backup media. But combined with mass storage technologies, it can serve quite well. For instance, by combining a high-speed network link to a remote data center containing large amounts of disk storage, suddenly the disadvantages about backing up to disks mentioned earlier are no longer disadvantages.

By backing up over the network, the disk drives are already off-site, so there is no need for transporting fragile disk drives anywhere. With sufficient network bandwidth, the speed advantage you can get from backing up to disk drives is maintained.

However, this approach still does nothing to address the matter of archival storage (though the same "spin off to tape after the backup" approach mentioned earlier can be used). In addition, the costs of a remote data center with a high-speed link to the main data center make this solution extremely expensive. But for the types of organizations that need the kind of features this solution can provide, it is a cost they gladly pay.

8.2.5. Storage of Backups

Once the backups are complete, what happens then? The obvious answer is that the backups must be stored. However, what is not so obvious is exactly what should be stored -- and where.

To answer these questions, we must first consider under what circumstances the backups are to be used. There are three main situations:

1. Small, ad-hoc restoration requests from users

2. Massive restorations to recover from a disaster

3. Archival storage unlikely to ever be used again

Unfortunately, there are irreconcilable differences between numbers 1 and 2. When a user accidentally deletes a file, they would like it back immediately. This implies that the backup media is no more than a few steps away from the system to which the data is to be restored.

In the case of a disaster that necessitates a complete restoration of one or more computers in your data center, if the disaster was physical in nature, whatever it was that destroyed your computers would also have destroyed the backups sitting a few steps away from the computers. This would be a very bad state of affairs.

Archival storage is less controversial; since the chances that it will ever be used for any purpose are rather low, if the backup media was located miles away from the data center there would be no real problem.

The approaches taken to resolve these differences vary according to the needs of the organization involved. One possible approach is to store several days worth of backups on-site; these backups are then taken to more secure off-site storage when newer daily backups are created.

Another approach would be to maintain two different pools of media:

- A data center pool used strictly for ad-hoc restoration requests
- An off-site pool used for off-site storage and disaster recovery

Of course, having two pools implies the need to run all backups twice or to make a copy of the backups. This can be done, but double backups can take too long, and copying requires multiple backup drives to process the copies (and probably a dedicated system to actually perform the copy).

The challenge for a system administrator is to strike a balance that adequately meets everyone's needs, while ensuring that the backups are available for the worst of situations.

8.2.6. Restoration Issues

While backups are a daily occurrence, restorations are normally a less frequent event. However, restorations are inevitable; they will be necessary, so it is best to be prepared.

The important thing to do is to look at the various restoration scenarios detailed throughout this section and determine ways to test your ability to actually carry them out. And keep in mind that the hardest one to test is also the most critical one.

8.2.6.1. Restoring From Bare Metal

The phrase "restoring from bare metal" is a system administrator's way of describing the process of restoring a complete system backup onto a computer with absolutely no data of any kind on it -- no operating system, no applications, nothing.

Overall, there are two basic approaches to bare metal restorations:

Reinstall, followed by restore

> Here the base operating system is installed just as if a brand-new computer were being initially set up. Once the operating system is in place and configured properly, the remaining disk drives can be partitioned and formatted, and all backups restored from backup media.

System recovery disks

> A system recovery disk is bootable media of some kind (often a CD-ROM) that contains a minimal system environment, able to perform most basic system administration tasks. The recovery environment contains the necessary utilities to partition and format disk drives, the device drivers necessary to access the backup device, and the software necessary to restore data from the backup media.

> **Note**
>
> Some computers have the ability to create bootable backup tapes and to actually boot from them to start the restoration process. However, this capability is not available to all computers. Most notably, computers based on the PC architecture do not lend themselves to this approach.

8.2.6.2. Testing Backups

Every type of backup should be tested on a periodic basis to make sure that data can be read from it. It is a fact that sometimes backups are performed that are, for one reason or another, unreadable. The unfortunate part in all this is that many times it is not realized until data has been lost and must be restored from backup.

The reasons for this can range from changes in tape drive head alignment, misconfigured backup software, and operator error. No matter what the cause, without periodic testing you cannot be sure that you are actually generating backups from which data can be restored at some later time.

8.3. Disaster Recovery

As a quick thought experiment, the next time you are in your data center, look around, and imagine for a moment that it is gone. And not just the computers. Imagine that the entire building no longer exists. Next, imagine that your job is to get as much of the work that was being done in the data center going in some fashion, some where, as soon as possible. What would you do?

By thinking about this, you have taken the first step of disaster recovery. Disaster recovery is the ability to recover from an event impacting the functioning of your organization's data center as quickly and completely as possible. The type of disaster may vary, but the end goal is always the same.

The steps involved in disaster recovery are numerous and wide-ranging. Here is a high-level overview of the process, along with key points to keep in mind.

8.3.1. Creating, Testing, and Implementing a Disaster Recovery Plan

A backup site is vital, but it is still useless without a disaster recovery plan. A disaster recovery plan dictates every facet of the disaster recovery process, including but not limited to:

- What events denote possible disasters
- What people in the organization have the authority to declare a disaster and thereby put the plan into effect
- The sequence of events necessary to prepare the backup site once a disaster has been declared
- The roles and responsibilities of all key personnel with respect to carrying out the plan
- An inventory of the necessary hardware and software required to restore production
- A schedule listing the personnel to staff the backup site, including a rotation schedule to support ongoing operations without burning out the disaster team members
- The sequence of events necessary to move operations from the backup site to the restored/new data center

Disaster recovery plans often fill multiple looseleaf binders. This level of detail is vital because in the event of an emergency, the plan may well be the only thing left from your previous data center (other than the last off-site backups, of course) to help you rebuild and restore operations.

Note

While disaster recovery plans should be readily available at your workplace, copies should also be stored off-site. This way, a disaster that destroys your workplace will not take every copy of the disaster recovery plan with it. A good place to store a copy is your off-site backup storage location. If it does not violate your organization's security policies, copies may also be kept in key team members' homes, ready for instant use.

Such an important document deserves serious thought (and possibly professional assistance to create).

And once such an important document is created, the knowledge it contains must be tested periodically. Testing a disaster recovery plan entails going through the actual steps of the plan: going to the backup site and setting up the temporary data center, running applications remotely, and resuming normal operations after the "disaster" is over. Most tests do not attempt to perform 100% of the tasks in the plan; instead a representative system and application is selected to be relocated to the backup site, put into production for a period of time, and returned to normal operation at the end of the test.

Note

Although it is an overused phrase, a disaster recovery plan must be a living document; as the data center changes, the plan must be updated to reflect those changes. In many ways, an out-of-date disaster recovery plan can be worse than no plan at all, so make it a point to have regular (quarterly, for example) reviews and updates of the plan.

8.3.2. Backup Sites: Cold, Warm, and Hot

One of the most important aspects of disaster recovery is to have a location from which the recovery can take place. This location is known as a *backup site*. In the event of a disaster, a backup site is where your data center will be recreated, and where you will operate from, for the length of the disaster.

There are three different types of backup sites:

- Cold backup sites
- Warm backup sites
- Hot backup sites

Obviously these terms do not refer to the temperature of the backup site. Instead, they refer to the effort required to begin operations at the backup site in the event of a disaster.

A cold backup site is little more than an appropriately configured space in a building. Everything required to restore service to your users must be procured and delivered to the site before the process of recovery can begin. As you can imagine, the delay going from a cold backup site to full operation can be substantial.

Cold backup sites are the least expensive sites.

A warm backup site is already stocked with hardware representing a reasonable facsimile of that found in your data center. To restore service, the last backups from your off-site storage facility must be delivered, and bare metal restoration completed, before the real work of recovery can begin.

Hot backup sites have a virtual mirror image of your current data center, with all systems configured and waiting only for the last backups of your user data from your off-site storage facility. As you can imagine, a hot backup site can often be brought up to full production in no more than a few hours.

A hot backup site is the most expensive approach to disaster recovery.

Backup sites can come from three different sources:

- Companies specializing in providing disaster recovery services
- Other locations owned and operated by your organization
- A mutual agreement with another organization to share data center facilities in the event of a disaster

Each approach has its good and bad points. For example, contracting with a disaster recovery firm often gives you access to professionals skilled in guiding organizations through the process of creating, testing, and implementing a disaster recovery plan. As you might imagine, these services do not come without cost.

Using space in another facility owned and operated by your organization can be essentially a zero-cost option, but stocking the backup site and maintaining its readiness is still an expensive proposition.

Crafting an agreement to share data centers with another organization can be extremely inexpensive, but long-term operations under such conditions are usually not possible, as the host's data center must still maintain their normal production, making the situation strained at best.

In the end, the selection of a backup site is a compromise between cost and your organization's need for the continuation of production.

8.3.3. Hardware and Software Availability

Your disaster recovery plan must include methods of procuring the necessary hardware and software for operations at the backup site. A professionally-managed backup site may already have everything you need (or you may need to arrange the procurement and delivery of specialized materials the site does not have available); on the other hand, a cold backup site means that a reliable source for every single item must be identified. Often organizations work with manufacturers to craft agreements for the speedy delivery of hardware and/or software in the event of a disaster.

8.3.4. Availability of Backups

When a disaster is declared, it is necessary to notify your off-site storage facility for two reasons:

- To have the last backups brought to the backup site
- To arrange regular backup pickup and dropoff to the backup site (in support of normal backups at the backup site)

Note

In the event of a disaster, the last backups you have from your old data center are vitally important. Consider having copies made before anything else is done, with the originals going back off-site as soon as possible.

8.3.5. Network Connectivity to the Backup Site

A data center is not of much use if it is totally disconnected from the rest of the organization that it serves. Depending on the disaster recovery plan and the nature of the disaster itself, your user community might be located miles away from the backup site. In these cases, good connectivity is vital to restoring production.

Another kind of connectivity to keep in mind is that of telephone connectivity. You must ensure that there are sufficient telephone lines available to handle all verbal communication with your users. What might have been a simple shout over a cubicle wall may now entail a long-distance telephone conversation; so plan on more telephone connectivity than might at first appear necessary.

8.3.6. Backup Site Staffing

The problem of staffing a backup site is multi-dimensional. One aspect of the problem is determining the staffing required to run the backup data center for as long as necessary. While a skeleton crew may be able to keep things going for a short period of time, as the disaster drags on more people will be required to maintain the effort needed to run under the extraordinary circumstances surrounding a disaster.

This includes ensuring that personnel have sufficient time off to unwind and possibly travel back to their homes. If the disaster was wide-ranging enough to affect peoples' homes and families, additional time must be allotted to allow them to manage their own disaster recovery. Temporary lodging near the backup site is necessary, along with the transportation required to get people to and from the backup site and their lodgings.

Often a disaster recovery plan includes on-site representative staff from all parts of the organization's user community. This depends on the ability of your organization to operate with a remote data center. If user representatives must work at the backup site, similar accommodations must be made available for them, as well.

8.3.7. Moving Back Toward Normalcy

Eventually, all disasters end. The disaster recovery plan must address this phase as well. The new data center must be outfitted with all the necessary hardware and software; while this phase often does not have the time-critical nature of the preparations made when the disaster was initially declared, backup sites cost money every day they are in use, so economic concerns dictate that the switchover take place as quickly as possible.

The last backups from the backup site must be made and delivered to the new data center. After they are restored onto the new hardware, production can be switched over to the new data center.

At this point the backup data center can be decommissioned, with the disposition of all temporary hardware dictated by the final section of the plan. Finally, a review of the plan's effectiveness is held, with any changes recommended by the reviewing committee integrated into an updated version of the plan.

8.4. Red Hat Enterprise Linux-Specific Information

There is little about the general topic of disasters and disaster recovery that has a direct bearing on any specific operating system. After all, the computers in a flooded data center will be inoperative whether they run Red Hat Enterprise Linux or some other operating system. However, there are parts of Red Hat Enterprise Linux that relate to certain specific aspects of disaster recovery; these are discussed in this section.

8.4.1. Software Support

As a software vendor, Red Hat, Inc does have a number of support offerings for its products, including Red Hat Enterprise Linux. You are using the most basic support tool right now by reading this manual. Documentation for Red Hat Enterprise Linux is available on the Red Hat Enterprise Linux Documentation CD (which can also be installed on your system for fast access), in printed form, and on the Red Hat website at *http://www.redhat.com/docs/*.

Self support options are available via the many mailing lists hosted by Red Hat (available at *https://www.redhat.com/mailman/listinfo*). These mailing lists take advantage of the combined knowledge of Red Hat's user community; in addition, many lists are monitored by Red Hat personnel, who contribute as time permits. Other resources are available from Red Hat's main support page at *http://www.redhat.com/apps/support/*.

More comprehensive support options exist; information on them can be found on the Red Hat website.

8.4.2. Backup Technologies

Red Hat Enterprise Linux comes with several different programs for backing up and restoring data. By themselves, these utility programs do not constitute a complete backup solution. However, they can be used as the nucleus of such a solution.

Note

As noted in *Section 8.2.6.1, “Restoring From Bare Metal”*, most computers based on the standard PC architecture do not possess the necessary functionality to boot directly from a backup tape. Consequently, Red Hat Enterprise Linux is not capable of performing a tape boot when running on such hardware.

However, it is also possible to use your Red Hat Enterprise Linux CD-ROM as a system recovery environment; for more information see the chapter on basic system recovery in the *System Administrators Guide*.

8.4.2.1. `tar`

The **`tar`** utility is well known among UNIX system administrators. It is the archiving method of choice for sharing ad-hoc bits of source code and files between systems. The **`tar`** implementation included with Red Hat Enterprise Linux is GNU **`tar`**, one of the more feature-rich **`tar`** implementations.

Using **`tar`**, backing up the contents of a directory can be as simple as issuing a command similar to the following:

```
tar cf /mnt/backup/home-backup.tar /home/
```

This command creates an archive file called **`home-backup.tar`** in **`/mnt/backup/`**. The archive contains the contents of the **`/home/`** directory.

The resulting archive file will be nearly as large as the data being backed up. Depending on the type of data being backed up, compressing the archive file can result in significant size reductions. The archive file can be compressed by adding a single option to the previous command:

```
tar czf /mnt/backup/home-backup.tar.gz /home/
```

The resulting **home-backup.tar.gz** archive file is now **gzip** compressed[5].

There are many other options to **tar**; to learn more about them, read the **tar(1)** man page.

8.4.2.2. cpio

The **cpio** utility is another traditional UNIX program. It is an excellent general-purpose program for moving data from one place to another and, as such, can serve well as a backup program.

The behavior of **cpio** is a bit different from **tar**. Unlike **tar**, **cpio** reads the names of the files it is to process via standard input. A common method of generating a list of files for **cpio** is to use programs such as **find** whose output is then piped to **cpio**:

```
find /home/ | cpio -o > /mnt/backup/home-backup.cpio
```

This command creates a **cpio** archive file (containing the everything in **/home/**) called **home-backup.cpio** and residing in the **/mnt/backup/** directory.

Note

Because **find** has a rich set of file selection tests, sophisticated backups can easily be created. For example, the following command performs a backup of only those files that have not been accessed within the past year:

```
find /home/ -atime +365 | cpio -o > /mnt/backup/home-backup.cpio
```

There are many other options to **cpio** (and **find**); to learn more about them read the **cpio(1)** and **find(1)** man pages.

8.4.2.3. dump/restore: Not Recommended for Mounted File Systems!

The **dump** and **restore** programs are Linux equivalents to the UNIX programs of the same name. As such, many system administrators with UNIX experience may feel that **dump** and **restore** are viable candidates for a good backup program under Red Hat Enterprise Linux. However, one method of using **dump** can cause problems. Here is Linus Torvald's comment on the subject:

```
From: Linus Torvalds To: Neil Conway Subject: Re: [PATCH] SMP race in ext2 - metadata
corruption. Date: Fri, 27 Apr 2001 09:59:46 -0700 (PDT) Cc: Kernel Mailing List <linux-
```

[5] The **.gz** extension is traditionally used to signify that the file has been compressed with **gzip**. Sometimes **.tar.gz** is shortened to **.tgz** to keep file names reasonably sized.

```
kernel At vger Dot kernel Dot org> [ linux-kernel added back as a cc ] On Fri, 27 Apr 2001,
 Neil Conway wrote: > > I'm surprised that dump is deprecated (by you at least ;-)). What
 to > use instead for backups on machines that can't umount disks regularly? Note that dump
 simply won't work reliably at all even in 2.4.x: the buffer cache and the page cache (where
 all the actual data is) are not coherent. This is only going to get even worse in 2.5.x,
 when the directories are moved into the page cache as well. So anybody who depends on "dump"
 getting backups right is already playing Russian roulette with their backups. It's not at
 all guaranteed to get the right results - you may end up having stale data in the buffer
 cache that ends up being "backed up". Dump was a stupid program in the first place. Leave
 it behind. > I've always thought "tar" was a bit undesirable (updates atimes or > ctimes for
 example). Right now, the cpio/tar/xxx solutions are definitely the best ones, and will work
 on multiple filesystems (another limitation of "dump"). Whatever problems they have, they
 are still better than the _guaranteed_(*) data corruptions of "dump". However, it may be
 that in the long run it would be advantageous to have a "filesystem maintenance interface"
 for doing things like backups and defragmentation.. Linus (*) Dump may work fine for you a
 thousand times. But it _will_ fail under the right circumstances. And there is nothing you
 can do about it.
```

Given this problem, the use of **dump**/**restore** on mounted file systems is strongly discouraged. However, **dump** was originally designed to backup unmounted file systems; therefore, in situations where it is possible to take a file system offline with **umount**, **dump** remains a viable backup technology.

8.4.2.4. The Advanced Maryland Automatic Network Disk Archiver (AMANDA)

AMANDA is a client/server based backup application produced by the University of Maryland. By having a client/server architecture, a single backup server (normally a fairly powerful system with a great deal of free space on fast disks and configured with the desired backup device) can back up many client systems, which need nothing more than the AMANDA client software.

This approach to backups makes a great deal of sense, as it concentrates those resources needed for backups in one system, instead of requiring additional hardware for every system requiring backup services. AMANDA's design also serves to centralize the administration of backups, making the system administrator's life that much easier.

The AMANDA server manages a pool of backup media and rotates usage through the pool in order to ensure that all backups are retained for the administrator-dictated retention period. All media is pre-formatted with data that allows AMANDA to detect whether the proper media is available or not. In addition, AMANDA can be interfaced with robotic media changing units, making it possible to completely automate backups.

AMANDA can use either **tar** or **dump** to do the actual backups (although under Red Hat Enterprise Linux using **tar** is preferable, due to the issues with **dump** raised in *Section 8.4.2.3, “**dump**/**restore**: Not Recommended for Mounted File Systems!”*). As such, AMANDA backups do not require AMANDA in order to restore files -- a decided plus.

In operation, AMANDA is normally scheduled to run once a day during the data center's backup window. The AMANDA server connects to the client systems and directs the clients to produce estimated sizes of the backups to be done. Once all the estimates are available, the server constructs a schedule, automatically determining the order in which systems are to be backed up.

Once the backups actually start, the data is sent over the network from the client to the server, where it is stored on a holding disk. Once a backup is complete, the server starts writing it out from the holding disk to the backup media. At the same time, other clients are sending their backups to the server for storage on the holding disk. This results in a continuous stream of data available for writing to the backup media. As backups are written to the backup media, they are deleted from the server's holding disk.

Once all backups have been completed, the system administrator is emailed a report outlining the status of the backups, making review easy and fast.

Should it be necessary to restore data, AMANDA contains a utility program that allows the operator to identify the file system, date, and file name(s). Once this is done, AMANDA identifies the correct backup media and then locates and restores the desired data. As stated earlier, AMANDA's design also makes it possible to restore data even without AMANDA's assistance, although identification of the correct media would be a slower, manual process.

This section has only touched upon the most basic AMANDA concepts. To do more research on AMANDA, start with the **amanda(8)** man page.

8.5. Additional Resources

This section includes various resources that can be used to learn more about disaster recovery and the Red Hat Enterprise Linux-specific subject matter discussed in this chapter.

8.5.1. Installed Documentation

The following resources are installed in the course of a typical Red Hat Enterprise Linux installation and can help you learn more about the subject matter discussed in this chapter.

- **tar(1)** man page -- Learn how to archive data.
- **dump(8)** man page -- Learn how to dump file system contents.
- **restore(8)** man page -- Learn how to retrieve file system contents saved by **dump**.
- **cpio(1)** man page -- Learn how to copy files to and from archives.
- **find(1)** man page -- Learn how to search for files.
- **amanda(8)** man page -- Learn more about the commands that are part of the AMANDA backup system.
- Files in **/usr/share/doc/amanda-server-*<version>*/** -- Learn more about AMANDA by reviewing these various documents and example files.

8.5.2. Useful Websites

- *http://www.redhat.com/apps/support/* -- The Red Hat support homepage provides easy access to various resources related to the support of Red Hat Enterprise Linux.
- *http://www.disasterplan.com/* -- An interesting page with links to many sites related to disaster recovery. Includes a sample disaster recovery plan.
- *http://web.mit.edu/security/www/isorecov.htm* -- The Massachusetts Institute of Technology Information Systems Business Continuity Planning homepage contains several informative links.
- *http://www.linux-backup.net/* -- An interesting overview of many backup-related issues.
- *http://www.linux-mag.com/1999-07/guru_01.html* -- A good article from Linux Magazine on the more technical aspects of producing backups under Linux.
- *http://www.amanda.org/* -- The Advanced Maryland Automatic Network Disk Archiver (AMANDA) homepage. Contains pointers to the various AMANDA-related mailing lists and other online resources.

8.5.3. Related Books

The following books discuss various issues related to disaster recovery, and are good resources for Red Hat Enterprise Linux system administrators:

- The *System Administrators Guide*; Red Hat, Inc -- Includes a chapter on system recovery, which could be useful in bare metal restorations.
- *Unix Backup &Recovery* by W. Curtis Preston; O'Reilly &Associates -- Although not written specifically for Linux systems, this book provides in-depth coverage into many backup-related issues, and even includes a chapter on disaster recovery.

Appendix A. Revision History

Revision 1.0	Tue Sep 23 2008	Don Domingo *ddomingo@redhat.com*

migrated to new automated build system

Index