The -h switch generates “human-readable” output, as seen in the df command described previously in this section. The -d switch causes du to output entries only at the given depth, with the current directory being 0, immediate subdirectories being 1, and so on. Use the -c switch to print a final, grand-total line. Also, instead of simply summing up the current directory, you can name the directory path, which in this case is named /Users.

Finally, Instruments.app is an ideal way to examine file activity and impact on a storage system for one or more processes. If df and du do not provide the information that you need, Instruments, with its capability to finely detail file and disk activity, and dtrace offer the necessary power and depth to provide that information. For more information on Instruments and dtrace, see the section “Instruments and DTrace” in Chapter 8, “Monitoring Systems.”

Evaluating the Upgrade History

When assessing a system, particularly a server, it is critical to know not only where it is now (the current operating system, load, hardware configuration, and so on), but also how the system got to where it is. Was the current operating system a clean installation? Or was it an upgrade? It is possible to figure this out, even if there is no prior system administrator around to ask. Without this knowledge, it is often difficult to correlate behavior that you see with baseline, known behavior.

Among other tasks, the Apple installer performs two actions when installing a package that can help you figure out the history of installed packages and system upgrades. First, the installer writes *entries* to the installer.log file, located in /var/log, along with several other log files. Second, the installer writes *receipts* to /Library/Receipts.

All instances of "installer.log" should read "install.log"

In the installer.log file, the installer program writes a running list of packages that it installs on the system. Other entries in installer.log are written by Software Update as it finds new software to install, and software update service, if the machine in question is running Mac OS X Server along with the software update daemon, swupd. An example of an initial system install from installer.log is as follows:

```
OSInstaller[197]: =========================================
OSInstaller[197]: Choices selected for installation:
OSInstaller[197]:   Install: "Mac OS X Server"
OSInstaller[197]:   Install: "Essential System Software"
OSInstaller[197]:       BaseSystem.pkg : com.apple.pkg.BaseSystem : 10.5.0.1.1.1192168948
```

All instances of "installer.log" should read "install.log"

For your purposes, the installer.log may have limited information. The system's periodic maintenance (in the daily folder at /etc/periodic/daily/600.daily.server) *rolls* logs—that is, compresses the current log file and starts a new one. Rolling entirely removes the oldest log files from a disk so that the log disk does not fill up. This means that if the server was installed or upgraded months ago, it is unlikely that a record of it will still exist in the installer.log files.

Receipts, on the other hand, do not expire and remain as a record of packages that have been installed.

The Apple installer, after installing the files that it contains (the payload of a component package), places a receipt in the /Library/Receipts directory of the installation volume. An *installation receipt* is a token that the installer uses to determine whether a package has already been installed on a system. If the installer, on subsequent installations of packages using the same package filename on the same volume, encounters a receipt, it processes the installation as an upgrade.

When the installer encounters a package in Mac OS X v10.5 format, Leopard handles receipts differently than earlier Macintosh operating systems. With earlier package formats, receipts were dropped by package name into /Library/Receipts. Each receipt resembled the original package minus the actual payload. The only way to remove receipts was manually.

Leopard, in contrast, drops receipts for v10.5-format packages into /Library/Receipts/boms (or Bill of Materials). Leopard also adds a new package database to the system, `/Library/Receipts/db`, which stores the receipts database. (You should not manipulate this database manually, or you risk corruption of its format.) Leopard also adds a command-line utility, `pkgutil`, to manipulate and query the database.

You can use `pkgutil` to collect information about a given package:

```
$ pkgutil --pkg-info com.apple.pkg.BaseSystem
package-id: com.apple.pkg.BaseSystem
version: 10.5.0.1.1.1192168948
volume: /
location: ./
install-time: 1208628236
groups: com.apple.repair-permissions.pkg-group com.apple.FindSystemFiles.pkg-group
```

What You've Learned

This chapter outlines tools that can assist you in evaluating or reassessing a system, including technical and nontechnical aspects. This chapter covered the following tools and techniques:

- How to determine hardware utilization, to assess the usage of storage, network, CPU, and memory
- Displaying vital statistics about a network interface using `netstat`, and from this information, manually computing utilization
- Displaying information about currently running system processes with the `ps` utility
- Computing load average as an important metric in determining CPU capacity
- Using `sysctl` as one way to show the current load average
- Displaying detailed statistics about virtual memory usage, including system average statistics since bootup, using `vm_stat`
- Displaying statistics about current and average input/output, using `iostat`
- Displaying disk capacity use with `df`
- Displaying disk usage for a given part of the disk hierarchy, such as /Users, with `du`
- Determining the history of system upgrades and installed packages using receipts and /var/log/installer.log install.log
- Querying and manipulating the receipts database using the `pkgutil` command
- Accounting of user workflows when assessing systems

Review Quiz

1. Which command-line utility supplies detailed statistics and information about a network interface?
2. What is the function of the `sysctl` command?
3. Which Mac OS X Server command reports detailed information about currently connected AFP and SMB users?
4. Name three ways to display the current load average.
5. Which command-line utility can quickly summarize disk capacity?