For example, the **mail** program (see the section "Sending Mail from a Shell Prompt" in Chapter 6) normally reads the message to send from your keyboard. Here's how to use the input redirection operator to mail the contents of the file *to_do* to *bigboss@corp.xyz*:

```
$ mail bigboss@corp.xyz < to_do
$
```

If a program writes to its standard output, which is normally the screen, you can make it write to a file instead by using the greater-than symbol (>) operator. The pipe operator (|) sends the standard output of one program to the standard input of another program. Input/output redirection is one of the most powerful and flexible Unix features, We'll take a closer look at it soon.

Putting Text in a File

Instead of always letting a program's output come to the screen, you can redirect output into a file. This is useful when you'd like to save program output or when you put files together to make a bigger file.

cat

cat, which is short for "concatenate," reads files and outputs their contents one after another, without stopping.

To display files on the standard output (your screen), use:

cat *file(s)*

For example, let's display the contents of the file */etc/passwd*. This system file describes users' accounts. (Your system may have a more complete list somewhere else.)

```
$ cat /etc/passwd
root:x&k8KP30f;(:0:0:Root:/:
daemon:*:1:1:Admin:/:
          .
          .
          .
john::128:50:John Doe:/usr/john:/bin/sh
$
```

You cannot go back to view the previous screens, as you can when you use a pager program such as **less** (unless you're using a terminal window with a scrollbar, that is). **cat** is mainly used with redirection, as we'll see in a moment.