

operator; so the output of `cat` goes to a file called *password* in the working directory. Displaying the file *password* shows that its contents are the same as the file */etc/passwd* (the effect is the same as the copy command `cp /etc/passwd password`).

You can use the `>` redirection operator with any program that sends text to its standard output—not just with `cat`. For example:

```
$ who > users
$ date > today
$ ls
password  today  users  ...
```

We've sent the output of `who` to a file called *users* and the output of `date` to the file named *today*. Listing the directory shows the two new files. Let's look at the output from the `who` and `date` programs by reading these two files with `cat`:

```
$ cat users
tim    tty1    Aug 12  07:30
john   tty4    Aug 12  08:26
$ cat today
Tue Aug 12 08:36:09 EDT 2001
$
```

You can also use the `cat` program and the `>` operator to make a small text file. We told you earlier to type `CTRL-D` if you accidentally enter `cat` without a filename. This is because the `cat` program alone takes whatever you type on the keyboard as input. Thus, the command:

```
cat > filename
```

takes input from the keyboard and redirects it to a file. Try the following example:

```
$ cat > to_do
Finish report by noon
Lunch with Xannie
Swim at 5:30
^D
$
```

`cat` takes the text that you typed as input (in this example, the three lines that begin with **Finish**, **Lunch**, and **Swim**), and the `>` operator redirects it to a file called *to_do*. Type `CTRL-D` *once*, on a new line by itself, to signal the end of the text. You should get a shell prompt.