By the way: if you enter **cat** without a filename, it tries to read from the keyboard (as we mention earlier). You can get out by pressing RETURN followed by a single CTRL-D .

The > operator

When you add "> *filename*" to the end of a command line, the program's output is diverted from the standard output to the named file. The > symbol is called the *output redirection operator.*

When you use the > operator, be careful not to accidentally overwrite a file's contents. Your system may let you redirect output to an existing file. If so, the old file will be deleted (or, in Unix lingo, "clobbered"). Be careful not to overwrite a much needed file!

Many shells can protect you from this risk. In the C shell, use the command **set noclobber**. The Korn shell and **bash** command is **set -o noclobber**. Enter the command at a shell prompt or put it in your shell's startup file. After that, the shell does not allow you to redirect onto an existing file and overwrite its contents.

This doesn't protect against overwriting by Unix programs such as **cp**; it works only with the > redirection operator. For more protection, you can set Unix file access permissions.

For example, let's use **cat** with this operator. The file contents that you'd normally see on the screen (from the standard output) are diverted into another file, which we'll then read using **cat** (without any redirection!):

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

An earlier example (in the section "cat") showed how **cat /etc/passwd** displays the file */etc/passwd* on the screen. The example here adds the >