

You can also create a bigger file from smaller files with the **cat** command and the **>** operator. The form:

```
cat file1 file2 > newfile
```

creates a file *newfile*, consisting of *file1* followed by *file2*.

```
$ cat today to_do > diary
$ cat diary
Tue Aug 12 08:36:09 EDT 2001
Finish report by noon
Lunch with Xannie
Swim at 5:30
$
```



You can't use redirection to add a file to itself, along with other files. For example, you might hope that the following command would merge today's to-do list with tomorrow's. This won't work!

```
$ cat to_do to_do.tomorrow > to_do.tomorrow
cat: to_do.tomorrow: input file is output file
```

cat warns you, but it's actually already too late. When you redirect a program's output to a file, Unix empties (clobbers) the file *before* the program starts running. The right way to do this is by using a temporary file (as you'll see in a later example) or simply by using a text editor program.

The >> operator

You can add more text to the end of an existing file, instead of replacing its contents, by using the **>>** (append redirection) operator. Use it as you would the **>** (output redirection) operator. So:

```
cat file2 >> file1
```

appends the contents of *file2* to the end of *file1*. For an example, let's append the contents of the file *users*, and also the current date and time, to the file *diary*. Then we display the file:

```
$ cat users >> diary
$ date >> diary
$ cat diary
Tue Aug 12 08:36:09 EDT 2001
```