Removing Files and Directories

You may have finished work on a file or directory and see no need to keep it, or the contents may be obsolete. Periodically removing unwanted files and directories frees storage space.

rm

The **rm** program removes files. The syntax is simple:

rm *filename(s)*

rm removes the named files, as the following example shows:

```
$ ls
chap10       chap2       chap5    cold
chap1a.old   chap3.old   chap6    haha
chap1b       chap4       chap7    oldjunk
$ rm *.old chap10
$ ls
chap1b    chap4    chap6    cold    oldjunk
chap2     chap5    chap7    haha
$ rm c*
$ ls
haha    oldjunk
$
```

When you use wildcards with **rm**, be sure you're deleting the right files! If you accidentally remove a file you need, you can't recover it unless you have a copy in another directory or in the system backups.

Do not enter **rm** * carelessly. It deletes all the files in your working directory.

Here's another easy mistake to make: you want to enter a command such as **rm c*** (remove all filenames starting with "c") but instead enter **rm c** * (remove the file named **c** and all files!).

It's good practice to list the files with **ls** before you remove them. Or, if you use **rm**'s **-i** (*i*nteractive) option, **rm** asks you whether you want to remove each file.

---

rmdir

Just as you can create new directories, you can remove them with the **rmdir** program. As a precaution, **rmdir** won't let you delete directories that