8.1 THE SHELL'S INTERPRETIVE CYCLE

When you log on to a UNIX machine, you first see a prompt. This prompt remains there until you key in something. Even though it may appear that the system is idling, a UNIX command is in fact running at the terminal. But this command is special; it's with you all the time and never terminates unless you log out. This command is the **shell.**

Even though the shell appears not to be doing anything meaningful when there's no activity at the terminal, it swings into action the moment you key in something. If you provide this input in the form of a **ps** command (that shows processes owned by you), you'll see the shell running:

```
$ ps
   PID TTY        TIME CMD                      Bash shell running
   328 pts/2      0:00 bash
```

The Bash shell is running at the terminal /dev/pts/2. When you key in a command, it goes as input to the shell. The shell first scans the command line for **metacharacters.** These are special characters that mean nothing to the command, but mean something special to the shell. The previous example (**ps**) had none of these characters, but we did encounter these characters before (section numbers shown in parentheses):

```
cat > foo            (5.1.2)
rm -r *              (5.3.1)
ls | more            (5.5.4)
comm file[12]        (5.11)
```

So what does the shell do when it sees metacharacters like the >, |, *, etc. in the command line? It performs all actions represented by the symbols *before* the command can be executed. For instance, the * makes no sense to **rm**, so the shell replaces it with all filenames in the current directory. **rm** ultimately runs with these filenames as arguments. The > means nothing to **cat**, so the shell creates the file foo and connects **cat**'s output to it.

When all pre-processing is complete, the shell passes on the command line to the kernel for ultimate execution. The command line now has none of the metacharacters that were originally seen by the shell. While the command is running, the shell has to wait for notice of its termination from the kernel. After the command has completed execution, the shell once again issues the prompt to take up your next command.

To summarize, the following activities are typically performed by the shell in its interpretive cycle:

- The shell issues the prompt and waits for you to enter a command.
- After a command is entered, the shell scans the command line for metacharacters and expands abbreviations (like the * in **rm** *) to recreate a simplified command line.
- It then passes on the command line to the kernel for execution.
- The shell waits for the command to complete and normally can't do any work while the command is running.
- After command execution is complete, the prompt reappears and the shell returns to its waiting role to start the next cycle. You are now free to enter another command.