

# Foreground and Background

You can launch any graphical Linux program from the shell. For example, you may run a text editor such as *kwrite* if you work in KDE or *gedit* if you use GNOME. Try this now. You'll notice that your shell becomes unusable once the editor has been launched. This is because the application has been started in the shell's foreground, which only has space for a single program. To regain access to the shell, press Ctrl-z. You'll see a message like this:

```
[1]+ Stopped kwrite
```

This indicates that *kwrite* has been stopped. The editor window no longer responds to mouse clicks and appears to be frozen. Now execute the command *bg* ("background"):

```
bg
```

This will continue execution of the editor program, relegating it to the shell's background. You can now use the shell while the editor is running, possibly starting more graphical programs. You should experiment with Ctrl-z and *bg* before you move on.

There is another key combination related to Ctrl-z that you'll use frequently: Ctrl-c. Pressing Ctrl-c terminates the process running in the foreground. Try to remember the following:

- Ctrl-z pauses a program
- Ctrl-c terminates a program

Run the command *jobs* to get a list of programs running in the shell's background:

```
jobs
```

The command also lists processes that have been stopped using Ctrl-z. The word "process" is a technical term for a program managed by Linux. The term "job" denotes a program that has been launched in the shell, which provides an additional layer of process management called "job control". Foregrounding and backgrounding are features of the shell's job control.

The command *fg* brings a program to the foreground. (You'll rarely need this.) When *bg* and *fg* are used without arguments, they relate to the job that the shell regards as current. If you want to foreground or background another job, specify the job's number on the command line, like so:

```
bg 3
```

This will background job no. 3 (if it exists). You can find out a job's number by calling *jobs*.

It is cumbersome to relegate programs to the background by pressing Ctrl-z and invoking *bg*. Often you'll want to start a program in the background right away. This is done by adding an ampersand (&) at the end of the command line, like this:

```
kwrite &
```

Take some time to experiment with the shell's job control mechanism. It's a powerful way of managing programs (akin to the "task bar" commonly found in graphical desktop environments), but it can also be confusing and dangerous if you're not used to it. You may lose data if you terminate a program by mistake.