When you close a shell session or logoff bash saves the last $HISTSIZE history lines in your home directory in a file called .bash\_history (~/.bash\_history). When you login again your history is loaded from this file. If you have multiple sessions open, then the history file is rewritten as each one closes. There is no merging of data from each session.

**Command history**

If you are typing in commands as you read, you may notice that you often use a command many times, either exactly the same or with slight changes. The good news is that the bash shell can maintain a *history* of your commands. By default, history is on. You can turn it off using the command set +o history and turn it back on using set -o history. An environment variable called HISTSIZE tells bash how many history lines to keep. A number of other settings control how history works and is managed. See the bash man pages for full details.

Some of the commands that you can use with the history facility are:

**history**

Displays the entire history.

**history *N***

Displays the last *N* lines of your history.

**history -d *N***

Deletes line *N* form your history; you might do this if the line contains a password, for example.

**!!**

Your most recent command.

**!*N***

The *N*th history command.

**!-*N***

The command that is *N* commands back in the history (!-1 is equivalent to !!).

**!#**

The current command you are typing.

**!*string***

The most recent command that starts with *string*.

**!?*string*?**

The most recent command that contains *string*.

You can also use a colon (:) followed by certain values to access or modify part or a command from your history. Listing 16 illustrates some of the history capabilities.

**Listing 16. Manipulating history**

[ian@atticf20 ~]$ **echo $$**

3175

[ian@atticf20 ~]$ **env -i bash -c 'echo $$'**

6737

[ian@atticf20 ~]$ **!!**

env -i bash -c 'echo $$'

6744

[ian@atticf20 ~]$ **!ec**

echo $$

3175

[ian@atticf20 ~]$ **!en:s/$$/$PPID/**

env -i bash -c 'echo $PPID'

3175

[ian@atticf20 ~]$ **history 6**

263 history -d259

264 echo $$

265 env -i bash -c 'echo $$'

266 echo $$

267 env -i bash -c 'echo $PPID'

268 history 6

[ian@atticf20 ~]$ **history -d266**

[ian@atticf20 ~]$ **!-2**

history 6

264 echo $$

265 env -i bash -c 'echo $$'

266 env -i bash -c 'echo $PPID'

267 history 6

268 history -d266

269 history 6

The commands in Listing 16 do the following:

1. Echo the current shell's PID.
2. Run an echo command in a new shell and echo that shell's PID.
3. Rerun the last command.
4. Rerun the last command starting with 'ec'; this reruns the first command in this example.
5. Rerun the last command starting with 'en', but substitute '$PPID' for '$$', so the parent PID is displayed instead.
6. Put a comment on the terminal output.
7. Display the last 6 commands of the history.
8. Delete history entry 266, the last echo command.
9. Redisplay the latest 6 commands of the history.

You can also edit the history interactively. The bash shell uses the readline library to manage command editing and history. By default, the keys and key combinations used to move through the history or edit lines are similar to those used in the GNU Emacs editor. Emacs keystroke combinations are usually expressed as **C-x** or **M-x**, where **x** is a regular key and **C** and **M** are the *Control* and *Meta* keys, respectively. On a typical PC system, the **Ctrl** key serves as the Emacs Control key, and the **Alt** key serves as the Meta key. Table 4 summarizes some of the history editing functions available. Besides the key combinations shown in Table 4, cursor movement keys such as the right, left, up, and down arrows, and the Home and End keys are usually set to work in a logical way. Additional functions, as well as how to customize these options using a readline init file (usually inputrc in your home directory), can be found in the man pages.