

grep (print lines matching a pattern)

This neat little program searches files for a regular expression pattern. It's called like this:

```
grep PATTERN FILE
```

If you don't know how to use regular expressions ("regexes" for short), you may want to have a look at the mini tutorial in Appendix A [see the [printed book](#)]. The following command will search the file `/var/log/messages` for the string "usb", followed by any number of characters (`.`), followed by the string "id":

```
grep `usb.*id` /var/log/messages
```

The command will output any lines that match the search pattern. I enclosed the pattern in inverted commas to make sure that the shell doesn't evaluate `*` as a file wildcard. It's a good habit to put inverted commas around regular expressions to ensure that they "get through" to `grep` without being interpreted by the shell.

Let me show you the most important `grep` options by way of example.

```
grep -i --color `\.jpe\?g` index
```

This will output any lines from `index` that contain the string `jpg` or `jpeg`. The `-i` option stands for "case-insensitive", which means that lines containing `JPG` or `JPEG` will be included in the output. The option `--color` highlights the expressions matched, making it easy to identify them in long output lines.

You'll often want to use `grep` to search through multiple files that are distributed across a directory tree. This can be done like so:

```
grep -l -R PATTERN .
```

This will search the current directory (`.`) for files containing `PATTERN`, recursing into any subdirectories (`-R`). The `-l` option suppresses `grep`'s normal output, giving you only a list of files where matches have been found. If you want to see the matched lines, leave out the option.