Chapter 2

Searching for fixed strings

Some jargon: A "string" is a commonly used technical expression for a sequence of characters.

2.1 The `grep` family of commands

If we are just searching for fixed strings, then there are three very similar commands available to us: `grep`, `egrep` and `fgrep`. Historically, `grep` came first and is named after the commands that have to be given to `sed` to make it search for strings. It was followed by `egrep` ("extended `grep`") which had much of `grep`'s functionality (but not all) and some more besides. `fgrep` was `grep` restricted to only looking for simple strings. The idea was that this would make it faster. (The leading "`f`" was meant to stand for "fast" as well as "fixed".) However, modern versions of `fgrep` are no faster than modern versions of `grep` and `egrep` so we will not concern ourselves with it in this course.

2.2 Searching for a string in a single file

First we will search for a simple string in a single file. In the examples directory (`~/text.tools.course`) there is a directory called `grep`. Within that are a couple of files, `virgin` and `wizard`. (In case you are interested, the former is a file detailing British Airways' attacks on Virgin Atlantic's computer records and the latter is a humorous discussion between a novice and a guru about the path to Unix enlightenment.)

```
$ cd
$ cd text.tools.course/grep-examples

$ ls
virgin  wizard

$ grep 'hack' virgin
For the next nine months the Helpline hackers
information they were hacking out of BABS. The
with a special card to prevent hacking.''
```

The command "`grep hack virgin`" has extracted each line from the file `virgin` that contained the string "`hack`". Note that the *string* "`hack`" was searched for, rather than the *word*. The two words that contained the string were "`hackers`" and "`hacking`". If the file had contained the word "`shack`", for instance, its line would have been printed too.

The single quotes around "`hack`" are not necessary in this particular case, but do no harm. In later examples the string being searched for will need quotes so it is a good habit to get into immediately. Their function is to protect any special characters in the search string from being processed by the shell.

Exercise 1: Search for the string "`hack`" in the file `wizard`.

2.3 Searching for a string in multiple files

To search more than one file, list the files after the search string, instead of just giving a single filename:

```
$ grep 'hack' virgin wizard
virgin:For the next nine months the Helpline hackers
virgin:information they were hacking out of BABS. The
virgin:with a special card to prevent hacking.''
wizard:of those who came before thee.  Hack, and hack
wizard:best hacks to the Net and gain in Repute
```

Note that when the `grep` command is searching more than one file it precedes each matching line with the filename of the file it found it in. This is particularly useful if you are trying to determine which of a set of files a string occurs in.

Exercise 2: Which file (out of `virgin` and `wizard`) contains the string "`limb`"?

To search every file in the current directory, the command to use is **$ grep wombat *** because the shell with expand the asterisk into a list of all the files.

2.4 Case insensitive searches

By default `grep` searches for strings case-sensitively, so it will not match the target string "`hack`" with a line containing "`Hack`" or "`HACK`". For example:

```
$ grep 'hack' wizard
of those who came before thee.  Hack, and hack
best hacks to the Net and gain in Repute

$ grep 'Hack' wizard
Hackerly Spirit in numerous pleasing ways), but
_The_New_Hacker's_Dictionary_ (Eric S. Raymond,
Student, but the Divine Foolery of the Hacker.
Hack, then; strive against Mighty Problems, have
Right-Thinking Hacker is to rewrite the UNIX
of those who came before thee.  Hack, and hack
until Hackers respect thy Power, and other
```

To search case-insensitively, use the `-i` option on `grep`:

```
$ grep -i 'hack' wizard
Hackerly Spirit in numerous pleasing ways), but
_The_New_Hacker's_Dictionary_ (Eric S. Raymond,
Student, but the Divine Foolery of the Hacker.
Hack, then; strive against Mighty Problems, have
Right-Thinking Hacker is to rewrite the UNIX
of those who came before thee.  Hack, and hack
best hacks to the Net and gain in Repute
until Hackers respect thy Power, and other

$ grep -i 'Hack' wizard
Hackerly Spirit in numerous pleasing ways), but
_The_New_Hacker's_Dictionary_ (Eric S. Raymond,
Student, but the Divine Foolery of the Hacker.
Hack, then; strive against Mighty Problems, have
Right-Thinking Hacker is to rewrite the UNIX
of those who came before thee.  Hack, and hack
best hacks to the Net and gain in Repute
until Hackers respect thy Power, and other
```

Note that even though the line

```
of those who came before thee.  Hack, and hack
```

matches in two places (as both "`Hack`" and "`hack`" match) the line is printed out only once.

Exercise 3: Search case insensitively for the string "`with`" in the two files, both individually and both at once.

2.5 Searching for words

As was demonstrated in section 2.2, grep searches for strings regardless of whether or not they form an entire word or are just part of a word. To cause grep to search for matching *words*, use the -w option:

```
$ grep 'hack' wizard
of those who came before thee.  Hack, and hack
best hacks to the Net and gain in Repute

$ grep -w 'hack' wizard
of those who came before thee.  Hack, and hack
```

The -w option can be used in conjunction with the -i option to search case-insensitively for a word:

```
$ grep -w 'hack' wizard
of those who came before thee.  Hack, and hack

$ grep -iw 'hack' wizard
Hack, then; strive against Mighty Problems, have
of those who came before thee.  Hack, and hack
```

Exercise 4: Demonstrate that the *string* "hack" appears in virgin but that the *word* does not.

2.6 Specifying the line number

As well as modifying the nature of grep's search, it is possible to modify the nature of its output. A common requirement is to know which line a match occurs on. To do this, add the -n option to the grep command:

```
$ grep 'hack' wizard
of those who came before thee.  Hack, and hack
best hacks to the Net and gain in Repute

$ grep -n 'hack' wizard
120:of those who came before thee.  Hack, and hack
122:best hacks to the Net and gain in Repute
```

If -n is applied when many files are being scanned, then the filename and the line number are printed:

```
$ grep 'hack' virgin wizard
virgin:For the next nine months the Helpline hackers
virgin:information they were hacking out of BABS. The
virgin:with a special card to prevent hacking.''
wizard:of those who came before thee.  Hack, and hack
wizard:best hacks to the Net and gain in Repute

$ grep -n 'hack' virgin wizard
virgin:44:For the next nine months the Helpline hackers
virgin:74:information they were hacking out of BABS. The
virgin:86:with a special card to prevent hacking.''
wizard:120:of those who came before thee.  Hack, and hack
wizard:122:best hacks to the Net and gain in Repute
```

2.7 Number of matches

If all you want is the number of matches in a file then the `-c` will suppress the generation of the matching lines and just give a count of the number of lines on which a match occured.

```
$ grep -iw 'hack' wizard
Hack, then; strive against Mighty Problems, have
of those who came before thee.  Hack, and hack

$ grep -iwc 'hack' wizard
2

$ grep -iwc 'hack' virgin wizard
virgin:0
wizard:2
```

Note that it is the number of *lines* with matches that is counted, not the number of actual matches. There is no mechanism for counting the number of matches.

Exercise 5: How many times does the word "`we`" occur in in the two files? And on how many lines?

2.8 Reversing the search

The final option on `grep` that we will cover is "`-v`". This causes the lines that do *not* match to be printed or to be counted:

```
$ grep -iwcv 'hack' *
virgin:94
wizard:150
```

Exercise 6: How many lines in the files do not contain the letter "`e`"?