【Linux Programming】Day10

Tags	
Date	@May 30, 2022
Summary	find and grep

【Ch2】Shell Programming

2.5 Commands(6)

The find command

The full syntax for the `find` command is as follows:

```
find [path] [options] [tests] [actions]
```

For example:

```
$ find / -mount -name test -print
/usr/bin/test
```

There are several options; the main ones are shown below:

Option	Meaning
`-depth`	Search the contents of a directory before looking at the directory itself.
`-follow`	Follow symbolic links.
`-maxdepths N`	Search at most *N* levels of the directory when searching.
`-mount` (or `-xdev`)	Don't search directories on other file systems.

For the tests. A large number of tests can be given to find, and each test returns either true of false. If the test returns false, then find stops considering the file and move on; if

the test returns true, then `find` processes the next `test` or `action` on the current file.

Test	Meaning
`-atime N`	The file was last accessed *N* days ago.
`-mtime N`	The file was last modified *N* days ago.
`-name pattern`	The name of the file, excluding any path, matches the pattern provided. To ensure that the pattern is passed to `find`, and not evaluated by the shell immediately, the pattern must always be in quotes.
`-newer otherfile`	The file is newer than the file `otherfile`.
`-type C`	The file is of type *C*, where *C* can be of a particular type; the most common are "d" for a directory and "f" for a regular file. For other types consult the manual pages.
`-user username`	The file is owned by the user with the given name.

We can also combine tests with operations:

Operator, Short Form	Operator, Long Form	Meaning
`!`	`-not`	Invert the test.
`-a`	`-and`	Both tests must be true.
`-o`	`-or`	Either test must be true.

Example:

Try searching in the current directory for files modified more recently than the file tmp.txt:

```
$ find . -newer tmp.txt -type f -print
```

Now, find files start either with a `c` or `a`.

```
$ find . \( -name "c*" -or -name "a*" \) -type f -print
```

The following is the most common actions:

Action	Meaning
`-exec command`	Execute a command. This is one of the most common actions. See the explanation following this table for how parameters may be passed to the command. This action must be terminated with a `\;` character pair.
`-ok command`	Like `-exec`, except that it prompts for user confirmation of each file on which it will carry out the command before executing the command. This action must be terminated with a `\;` character pair.
`-print`	Print out the name of the file.
`-ls`	Use the command `ls -dils` on the current file.

The `-exec` and `-ok` commands take subsequent parameters on the line as part of their parameters, until terminated with a `\;`.

```
$ find . -newer while2 -type f -exec ls -l {} \;
```

The grep Command

`grep` stands for general regular expression parser.

We use find to search our system for files, but we use `grep` to search files for strings.

The `grep` command takes options, a pattern to match, and files to search in:

```
grep [options] PATTERN [FILES]
```

If no filenames are given, it searches the standard input.

Option	Meaning
`-c`	Rather than print matching lines, print a count of the number of lines that match.
`-E`	Turn on extended expressions.
`-h`	Suppress the normal prefixing of each output line with the name of the file it was found in.
`-i`	Ignore case.
`-l`	List the names of the files with matching lines; don't output the actual matched line.
`-v`	Invert the matching pattern to select nonmatching lines, rather than matching lines.

```
$ grep in words.txt
When shall we three meet again. In thunder, lightning, or in rain?
I come, Graymalkin!
$ grep -c in words.txt words2.txt
words.txt:2
words2.txt:14
$ grep -c -v in words.txt words2.txt
words.txt:9
words2.txt:16
```

Regular Expressions

Character	Meaning
`^`	Anchor to the beginning of a line
`$`	Anchor to the end of a line
`.`	Any single character
`[ ]`	The square braces contain a range of characters, any one of which may be matched, such as a range of characters like a–e or an inverted range by preceding the range with a ^ symbol.

Match Pattern	Meaning
`[:alnum:]`	Alphanumeric characters
`[:alpha:]`	Letters
`[:ascii:]`	ASCII characters
`[:blank:]`	Space or tab
`[:cntrl:]`	ASCII control characters
`[:digit:]`	Digits
`[:graph:]`	Noncontrol, nonspace characters
`[:lower:]`	Lowercase letters
`[:print:]`	Printable characters
`[:punct:]`	Punctuation characters
`[:space:]`	Whitespace characters, including vertical tab
`[:upper:]`	Uppercase letters
`[:xdigit:]`	Hexadecimal digits

Option	Meaning
`?`	Match is optional but may be matched at most once
`*`	Must be matched zero or more times
`+`	Must be matched one or more times
`{n}`	Must be matched *n* times
`{n,}`	Must be matched *n* or more times
`{n,m}`	Must be matched between *n* or *m* times, inclusive

1. Start by looking for lines that end with the letter *e*. You can probably guess you need to use the special character `$`:

```
$ grep  e$ words2.txt
Art thou not, fatal vision, sensible
I see thee yet, in form as palpable
Nature seems dead, and wicked dreams abuse
$
```

As you can see, this finds lines that end in the letter *e*.

2. Now suppose you want to find words that end with the letter *a*. To do this, you need to use the special match characters in braces. In this case, you use `[[:blank:]]`, which tests for a space or a tab:

```
$ grep a[[:blank:]] words2.txt
Is this a dagger which I see before me,
A dagger of the mind, a false creation,
Moves like a ghost. Thou sure and firm-set earth,
$
```

3. Now look for three-letter words that start with *Th*. In this case, you need both `[[:space:]]` to delimit the end of the word and `.` to match a single additional character:

```
$ grep Th.[[:space:]] words2.txt
The handle toward my hand? Come, let me clutch thee.
The curtain'd sleep; witchcraft celebrates
Thy very stones prate of my whereabout,
$
```

4. Finally, use the extended `grep` mode to search for lowercase words that are exactly 10 characters long. Do this by specifying a range of characters to match *a* to `z`, and a repetition of 10 matches:

```
$ grep -E [a-z]\{10\} words2.txt
Proceeding from the heat-oppressed brain?
And such an instrument I was to use.
The curtain'd sleep; witchcraft celebrates
Thy very stones prate of my whereabout,
$
```