Using GREP

2016

Grep is a super useful tool, but it is difficult to master, so in this section we are going to explore some examples of the use of GREP. Grep can be as complex as your imagination can muster, but some very simple examples of Grep use will get you started nicely.

The name, "grep", derives from the command used to perform a similar operation, using the Unix/Linux text editor ed: g/re/p

grep command syntax

The syntax is as follows:

```
grep 'word' filename
grep 'word' file1 file2 file3
grep 'string1 string2' filename
cat otherfile | grep 'something'
command | grep 'something'
command option1 | grep 'data'
grep --color 'data' filename
```

How do I use grep command to search a file?

Search /etc/passwd file for boo user, enter:

```
grep pi /etc/passwd
returns: pi:x:1000:1000:,,,:/home/pi:/bin/bash (the two instanced of the word 'pi' will be coloured.
```

You can force grep to ignore word case i.e match boo, Boo, BOO and all other combination with the -i option:

```
grep -i "pi" /etc/passwd
This will return the word pi, PI, pI or Pi (all case combinations are allowed)
```

Use grep recursively

You can search recursively i.e. read all files under each directory for a string "192.168.1.5"

```
grep -r "192.168.1.5" /etc/
or
grep -R "192.168.1.5" /etc/
example return: /etc/dhcpcd.conf:static ip_address=192.168.1.5/24
```

Use grep to search words only

When you search for boo, grep will match fooboo, boo123, barfoo35 and more. You can force the grep command to select only those lines containing matches that form whole words i.e. match only boo word:

```
grep -w "boo" file
```

Use grep to search 2 different words

Use the egrep command as follows:

```
egrep -w 'word1|word2' /path/to/file
```

Count line when words has been matched

The grep can report the number of times that the pattern has been matched for each file using -c (count) option:

```
grep -c 'word' /path/to/file
```

Pass the -n option to precede each line of output with the number of the line in the text file from which it was obtained:

```
grep -n 'bash' /etc/passwd
example:
1:root:x:0:0:root:/root:/bin/bash
23:pi:x:1000:1000:,,,:/home/pi:/bin/bash
The word 'bash' was found on lines 1 and 23
```

Grep invert match

You can use -v option to print inverts the match; that is, it matches only those lines that do not contain the given word. For example print all line that do not contain the word bar:

```
grep -v bar /path/to/file
```

UNIX / Linux pipes and grep command

grep command often used with shell pipes. In this example, show the name of the hard disk devices:

```
dmesg | egrep '(s|h)d[a-z]'
[    2.535530] sdhci: Secure Digital Host Controller Interface driver
[    2.539697] sdhci: Copyright(c) Pierre Ossman
[    2.542163] sdhost: log_buf @ bac46000 (fac46000)
[    2.619698] mmc0: sdhost-bcm2835 loaded - DMA enabled (>1)
[    2.691142] sdhci-pltfm: SDHCI platform and OF driver helper
[    6.716496] bcm2835-wdt 3f100000.watchdog: Broadcom BCM2835 watchdog timer
[    7.186573] brcmfmac: brcmf_sdio_drivestrengthinit: No SDIO Drive strength init done for chip 43430
rev 1 pmurev 24
```

Display cpu model name:

```
cat /proc/cpuinfo | grep -i 'Model'
model name      : ARMv7 Processor rev 4 (v7l)
model name      : ARMv7 Processor rev 4 (v7l)
model name      : ARMv7 Processor rev 4 (v7l)
model name      : ARMv7 Processor rev 4 (v7l)
```

However, above command can be also used as follows without shell pipe:

```
grep -i 'Model' /proc/cpuinfo
```

How do I list just the names of matching files?

Use the -l option to list file name whose contents mention main():

```
grep -l 'main' *.c
```

Force grep to display output in colours

Finally, you can force grep to display output in colors, enter:
Raspbian seems to do this by default

```
grep --color pi /etc/passwd
```

From:
http://cameraangle.co.uk/ - **WalkerWiki - wiki.alanwalker.uk**

Permanent link:
http://cameraangle.co.uk/doku.php?id=using_grep

Last update: **2023/03/09 22:35**