Everything You Ever (Never?) Wanted to Know About Grep

(6/2006, -Mark A. Davis/TWUUG)

Intro: Grep is the Unix file pattern-searching command. Probably one of the most useful commands any Unix user should know. Grep will search files or standard input, looking for matches of things you specify and showing you the matching lines that it finds.

History: The name comes from a command in the ancient Unix text editor, “ed”, g/re/p which means "search **G**lobally for lines matching the **R**egular **E**xpression, and **P**rint them". Grep appeared in Unix in 1973. McIlroy asked Thompson (one of the fathers of Unix) to create a program to help him with some work that he was trying to do:

> "One afternoon I asked Ken Thompson if he could lift the regular expression recognizer out of the editor and make a one pass program to do it. He said yes. The next morning I found a note in my mail announcing a program named grep. It worked like a charm. When asked what that funny name meant, Ken said it was obvious. It stood for the editor command that it simulated, g/re/p (global regular expression print)."

What McIlroy refers to as "the unarticulated notion of software tools...brought home by the liberation of...grep from within the editor" had become one of the significant achievements of the Bell Labs research work on Unix. Researchers discovered that they could create a plethora of software tools to be used in varying combinations, thus facilitating the customized application by a user, a goal sought by those doing research in programming. Thus, the authors realized and created one of the philosophies of Unix- that it should be composed of many small tools that do one thing really well and can be combined together in any number of ways to get work done efficiently.

Other Greps: A long time ago, there was also an “fgrep” command, which meant “fast grep” because it didn't use regular expressions and “egrep”, which meant “extended grep” used with extended regular expressions. Both are now obsolete in gnu grep, and are just symbolic links to “grep”.

Grep Syntax: grep [options] PATTERN [FILE...] The simplest way to use grep is to just specify a search pattern and a filename like this:

grep twuug * This would search all files in the current directory for the word “twuug” and print all the matching lines to the screen (standard output). It will also tell you the filename (first), if it is searching more than one file.

Options: Grep has lots of options. These are the most common and useful:

-a, –text	Process binary files as text files. This is useful when grep/magic thinks a file is binary when it really isn't. Searching binary files (non-text) normally requires a binary editor.
-c, –count	Don't show all the matching lines, instead, just print a count of the matches.
-i, –ignore-case	Ignore upper/lowercase. In grep, like most things in Unix, it is case sensitive, so “UNIX” will not match “Unix” or “unix”.
-l, –files-with-matches	Only list the filename that contains matches, don't show the match itself.
-q, –quiet	Don't print anything on the standard output. This is useful if you are using grep in a script with “if grep” to test to see if something is true or false.
-R,-r,–recursive	Search recursively, searching all the files in all subdirectories.
-v, –invert-match	Instead of showing lines or files that MATCH the pattern, invert the process and show only lines and files that DO NOT match the given search pattern.

-w, –word-regexp	Assume the search key is a full word and match ONLY full words. This will prevent partial word matches. For example, if the search key is “cat”, using -w will prevent matching “category”, “catacomb”, and “meercat”.

Expressions: In addition to just words, you can use regular expressions as search keys. Here are some useful examples (they can be combined and there are MANY other expressions possible):

^cat	Search for “cat”, but only at the beginning of a line
cat$	Search for “cat”, but only at the end of a line
c.t	Search for anything that starts with “c”, ends with “t” and has one other character in the middle.
cat|rabbit	Search for “cat” OR “rabbit”. Note that you need to protect this expression from the shell with apostrophes (‘).
c[aoi]t	Any one of the characters in brackets. This would match “cat”, “cot”, or “cit”. You can also use a range of characters with a dash. For example, [a-d] would match the single character of a, b, c, or d.

Practical Uses of Grep: grep is very useful in scripts in addition to just using it on the command line and also when using it with pipes from the output of other commands. When used with pipes, it acts as a convenient filter, giving you only stuff you are interested in seeing. This is especially useful for long log files.

grep -i kernel /var/log/messages # show anything in the logs that has anything do with the kernel

who | grep mark # see if mark is logged in

last mark | grep toad # show all login sessions for mark but only from computer named toad

ls -l /usr/bin | grep Aug # find files modified in Aug of any year in /usr/bin

grep -c twuug.org /var/log/Mail/info # give a count of how many messages from TWUUG

grep Invoice listing.csv > listing2.csv # build a new file but only with invoice entries in it

ps -elf | grep netscape # show all netscape processes running on the system

grep “:Food” /etc/passwd | sort # show all users in Food Services, sorted by login name

grep -iR error $HOME # search all files in my home dir structure looking for “error” (upper or lower)

grep -cw mark /etc/group # tell how many groups user “mark” is in.

```
# Find only files modified in the last day in /doc and print only the names of those files that contain the word
# “config” in any case (upper or lower).
for FILE in ‘find /doc -mtime -1‘ ; do grep -li config $FILE ; done

#Check if mark is logged in and take action on that fact
while true
 do if who | grep -q ^mark” “
     then echo “Mark is logged in!”
     else echo “Sorry, no Mark yet :(“
    fi
 sleep 10
done
```

Colorful Grep: Since grep usually returns the entire matching lines of what it finds, it is sometimes hard to tell exactly what was found. There is a useful color option in gnu grep that will highlight the matches on the output. Just use the “–color=always” option. By default, all the matches will be highlighted in red!