

CS-333 - Unix Operating System Fundamentals

Lecture 9 Using grep & sed

Based on slides created by
Dr. Bangalore for the
Spring 2005 offering of
the course

grep

- grep = Get (or Global) Regular Expression and Print
- Filters out lines that do not meet given criteria
- Besides **grep**, we have **egrep** and **fgrep**
- Main difference is in the type of text pattern each of them can filter
- We'll first look at common features of three versions and then look at the differences

10/27/2006

3

grep

- To search through several files
 - `grep pattern file1 file2`
 - `grep pattern file*`
- To list only the filename that match
 - `grep -l pattern files`
- To count the number of matches
 - `grep -c pattern files`
- To search all files in a directory tree
 - `grep -r pattern directory` (not on Solaris)
- To match complete lines
 - `grep -x 'pattern' files` (not on Solaris)
- To search for patterns starting with -
 - `grep -e -pattern files` (not on Solaris)

10/27/2006

4

grep options in short

- `-h` -> omit name of the file from output during multi file search
- `-n` -> precede matching line with its line number
- `-i` -> ignore case
- `-l` -> display files that contain given pattern
- `-v` -> display lines that do NOT match the given pattern
- `-w` -> restrict matching to only whole words

10/27/2006

5

Regular Expressions

Metacharacter	Meaning
.	Matches any single character
[]	Matches any single character enclosed in brackets. A hyphen may be used to represent a range of characters. If the first character after the [is ^, then any character <i>not</i> enclosed in the brackets is matched. The *, ^, \$, and \ metacharacters lose their special meaning when used inside brackets.
*	May follow any character and denotes zero or more occurrences of the character that precedes it.
^	Matches the beginning of a line only.
\$	Matches the end of the line only.
\	The meaning of any metacharacter may be inhibited by preceding it with \.

10/27/2006

Source: "UNIX for Programmers and Users" by G. Glass & K. Abigs

Using metacharacters with grep (I)

- Selecting lines with a pattern at the beginning of the line
 - `grep '^pattern' files`
- Selecting lines with a pattern at the end of the line
 - `grep 'pattern$' files`
- Selecting lines that match alternative characters
 - `grep '[abc]pattern[xyz]' files` (match line that starts with either *a* or *b* or *c*, followed by *pattern* and ending with *x* or *y* or *z*)

10/27/2006

7

Using metacharacters with grep (II)

- Using metacharacter "." to match any single character
 - `grep '.' files`
 - `grep '^...$' files` (match line with only any three single characters on it)
- Locating a word explicitly
 - `grep -w word files`
 - `grep '\<word\>' files`
- Locating a range of characters
 - `grep '[A-Z0-9a-z]' files` (match any line that has a single upper, lower case letter, or a digit)
 - `grep '^ [A-Z][0-9][a-z]' files` (match any line starting with an upper case letter, followed by a digit and then a lower case letter)
 - `grep '^ [^a-z]' files` (lines in *files* that do not begin with lower case *a* through *z*)

10/27/2006

8

Using metacharacters with grep (III)

- Searching for character repetitions
 - `grep 'a*' files` (zero or more *a* characters)
- Locating lines that contain any sequence of characters
 - `grep 'a.*b' files` (*a* followed by zero or more characters, followed by *b*)
- All these different options can be combined to form a complex expression
 - `grep '^ [^0-9][0-9][A-Z]*$' files` (look for lines that start with not a digit, followed by a digit, zero or more uppercase letters, and ends with end-of-line)

10/27/2006

9

Fast grep - fgrep

- Searches for a *string* instead of searching for a pattern that matches an expression
- Uses a fast and compact algorithm
- No metacharacter expansion is performed
- We can obtain the functionality of fgrep with grep by using the *-F* option
- fgrep also accepts multiple patterns by reading a file with different patterns
 - `fgrep -f patternsfile files`

10/27/2006

10

Extended Regular Expressions

Metacharacter	Meaning
+	Matches one or more occurrences of the single preceding character.
?	Matches zero or one occurrences of the single preceding character.
	If you place a pipe symbol between two regular expressions it acts as an "or" operator.
()	If you place a regular expression in parentheses, you may use *, +, or ? metacharacters to operate on the entire expression rather than just one character.

Source: "UNIX for Programmers and Users" by G. Glass & K. Ables

10/27/2006

11

Extended grep – egrep

- Uses full regular expressions to match the patterns (could be slower than grep)
 - `egrep '^a|b$' files` (starts with a OR ends with b)
 - `egrep -f patternsFile files`
- To specify one or more of a previous character
 - `egrep 'ab+' files` (*a* followed by one or more *b*)
 - `egrep 'ab*' files` (*a* followed by zero or more *b*'s)
- To specify optional character use ?
 - `egrep 'a?b' files` (zero or one *a* followed by *b*)
- To specify number of character to match
 - `egrep 'a{3}b' files` (three *a*'s followed by a *b*)

10/27/2006

12

Using “sed” (I)

- To quit after matching a specified line number
`sed '20 q' filename` (first 20 lines are printed)
- To quit after the first matching of a given pattern
`sed '/pattern/ q' filename`
- To delete specific lines
`sed '1,10 d' filename` (lines 1-10 are deleted)
- To delete lines that match a pattern
`sed '/pattern/ d' filename`
Examples:
 - `sed '/xyz/ d' filename`
 - `sed '3,/abc/ d' filename`

10/27/2006

13

Using “sed” (II)

- Regular expressions can also be used as patterns for quit and delete
 - `sed '/^[A-Z].*[0-9]$/ q' filename` (quit printing once a line starting with upper case letter and ending with a digit is found, with any number of any character in between)
 - `sed '/^$/ d' filename` (delete blank lines)
- To replace the first instance of a specified pattern
`sed 's/abc/ABC/' filename`
- To replace multiple instances of a specified pattern
`sed 's/abc/ABC/g' filename`

10/27/2006

14

Using “sed” (III)

- To suppress output from sed use `-n` option
 - `sed -n '/pattern/ =' filename` (only the first matching line number is displayed)
 - `sed -n '3,6 p' filename` (print lines 3 through 6, without `-n` option all lines will be printed once while lines 3-6 will be printed twice)
 - `sed -n '1,15 s/abc/ABC/ p' filename` (the first line between 1 and 15 that matches the pattern will be printed - `abc` will be replaced with `ABC` too) **[Linux Only]**
- To print lines at intervals
 - `sed -n '1~2 p' filename` (starting with line 1, output every other line)
 - `sed -n '2~3 p' filename` (starting with line 2, output every third line)

10/27/2006

15

Using “sed” (IV)

- To print lines that do not match use `!p`
 - `sed -n '/^[0-9]/ !p' filename` (don't print lines starting with a number)
 - `sed -n '/^$/ !p' filename` (don't print blank lines)
- Complex substitution using contextual addresses **[Linux Only]**
 - `sed -n '/pattern1/ s/pattern2/replace/g p' filename` (match line with `pattern1` and then within matched line, match `pattern2` and replace it with `replace` string)
 - `sed -n '/[1-5]/ s/abc/ABC/g p' filename`
 - `sed -n '5,/abc/ s/xyz/XYZ/' filename`

10/27/2006

Address 1 Address 2

16

Reading and Writing Files from sed

- To read in a file at a specified location
 - `sed '5 r inputfile' filename` (insert `inputfile` after line 5 in file `filename` - if 5 was omitted, contents of `inputfile` would be added after each line in `filename`)
 - `sed '/abc/ r inputfile' filename` (after each line that contains `abc` insert the file `inputfile`)
- To write specific lines to another file
 - `sed '1,10 w outfile' filename` (write lines 1 through 10 from file `filename` to the file `outfile`)
 - `sed '/^[a-zA-Z]/ w outfile' filename` (write lines that start with a character to the file `outfile`)

10/27/2006

17

Passing Multiple Instructions to sed

- To pass multiple instructions to sed on the command line use `-e` option
 - `sed -e 's/abc/ABC/' -e 's/xyz/XYZ/' filename`
 - `sed -e '/^[a-z]/ p' -e 's/[0-9]/XXX/g p' filename` (print any line starting with lower case letter (these lines will be printed twice) and also replace any digit with `XXX`)
- We can also include the different instructions in a separate file and specify this commands file as input to sed with `-f` option
 - `sed -f optscript filename`

```
$ cat > optscript
s/root/ROOT/g
s/\bin/vcsh/\bin/vtsh/g
$ cat /etc/passwd | sed -f optscript
```

10/27/2006

18

Insert/Append Text with sed

- To insert text before a specified line use
 - `sed '5 i some text' filename`
 - `sed '/abc/i some text' filename` (insert *some text* before the line where *abc* is matched)
- To insert text after a specified line use
 - `sed '5 a some text' filename`
 - `sed '/abc/a some text' filename`
- When using a script file as input to sed, the script file should have
 - `sed -f scriptfile filename`
Where *scriptfile* has the following format:
`/abc/\`
`Text To Be Inserted Before The Line`
`/abc/a\`
`Text To Be Appended After The Line`

10/27/2006

19