

LPIC-1 101-400 – Lesson 7

103.7 Search text files using regular expressions



Regular Expressions

- Regular expressions is a versatile method to match strings and text patterns
- Regular expressions are part of the POSIX standard for UNIX systems
- There are two flavors, the "Basic Regular Extensions" and the "Extended Regular Expressions"



Regular Expressions Special Characters

Basic Regular Expressions	Extended Regular Expressions	Description
.	.	Matches any single character e.g. a.c matches with a1c , aac , abc , a)c etc
[]	[]	Matches any single character enclosed in the square brackets e.g. [agk] , [1-9a-kL-Z] . Here a "." and other special characters are interpreted literally
[^]	[^]	Matches any character NOT included in square brackets e.g. [^agk]
^	^	Matches the beginning of string or line e.g. ^hat , ^[jtr]
\< \>	\< \>	Matches words
\$	\$	Matches the end of string or line e.g. hat\$, [jtr]\$
*	*	Matches 0 or more of the preceding characters e.g. a* , [jtr]* , .*
\?	\?	Matches 0 or 1 of the preceding characters e.g. a? , [jtr]??
\+	\+	Matches 1 or more of the preceding characters e.g. a+ , [jtr]+
\	\	Matches the expression before or after the vertical bar e.g. this that
\{m,n\}	\{m,n\}	Matches m times the preceding characters but no more than m (n>m) e.g. a{2,4} , [jtr]{2,4}
\(\)	\(\)	Grouping of the expression in parentheses so it can be called later. \1 refers to the first set of parentheses, \2 to the second, and goes up to \9

Character Classes in Regular Expressions

Character Classes	Equivalent Expression	Description
<code>[:alnum:]</code>	<code>[a-zA-Z0-9]</code>	Alphanumeric characters
<code>[:alpha:]</code>	<code>[a-zA-Z]</code>	Alphabetic characters
<code>[:blank:]</code>	<code>[\t]</code>	Space and Tab
<code>[:cntrl:]</code>	<code>[\x00-\x1F\x7F]</code>	Control characters
<code>[:digit:]</code>	<code>[0-9]</code>	Numbers
<code>[:graph:]</code>	<code>[\x21-\x7E]</code>	Optical characters
<code>[:lower:]</code>	<code>[a-z]</code>	Lower case letters
<code>[:print:]</code>	<code>[\x20-\x7E]</code>	All optical characters plus space
<code>[:punct:]</code>	<code>[\]\[!\"#\$%&'()*+,-./:;<=>?@\^_`{ }~-]</code>	Punctuation characters
<code>[:space:]</code>	<code>[\t\r\n\v\f]</code>	All Whitespace
<code>[:upper:]</code>	<code>[A-Z]</code>	Upper case letters
<code>[:xdigit:]</code>	<code>[A-Fa-f0-9]</code>	Hexadecimal digits

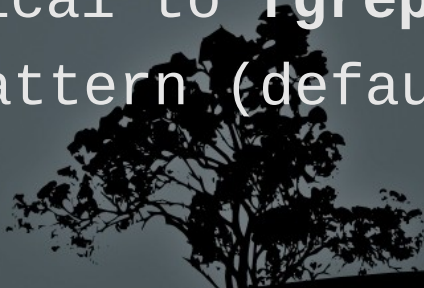
Match Expressions with `grep`

- **grep** is one of the most powerful and popular commands for text filtering, in Linux
- It prints lines that match a certain pattern, from text files
- Support Basic Regular Expressions
- Supports extended regular expressions when the **-E** option is set (**grep -E = egrep**)
- It can even search for patterns in binary files and report the matching ones



Match Expressions with `grep`

Options:

- `-c` # just show the number of matches in the text
 - `-i` # ignores upper case or lower case in the match
 - `-n` # show the number of line of the match, in the text
 - `-o` # show only the match, not the complete line
 - `-r` # recursively search the directory hierarchy
 - `-v` # show the lines NOT matching a match
 - `-E` # support Extended Regular Expressions. Identical to `egrep`)
 - `-F` # literal interpretation of the special characters in the pattern. Identical to `fgrep`
 - `--color=auto` # color the matched pattern (default in Ubuntu)
- 

Match Expressions with `grep`

- `$ grep uuid /etc/fstab # regular expressions are case sensitive`
- `$ grep UUID /etc/fstab # does it work now?`
- `$ grep -i uuid /etc/fstab # case insensitive search`
- `$ grep -v -i uuid /etc/fstab # show lines that do not include uuid`
- `$ grep -i "[linux]\{3,5\}" -r /etc/ # recursively search /etc for patterns of the enclosed characters, which are equal or greater than 3 up to equal or less than 5`
- `$ grep -i "[linux]\{5\}" -r /etc/ # search for patterns with exactly 5 characters from those enclosed in the brackets`

Match Expressions with `grep`

- `$ grep "^[[:alpha:]]" -r /etc#` print lines beginning with letters.
- `$ grep "\<[[:alnum:]]*\>" -r /etc #` print lines beginning with letters or numbers
- `$ grep "[[:punct:]]" -r /etc #` print lines containing punctuation
- `$ grep "[[:space:]]$" -r /etc #` print lines that end in whitespace
- `$ grep "[[:xdigit:]]\{3,\}" -r /etc #` print lines that contain 3 or more hexadecimal digits



Variations of `grep`, `egrep` and `fgrep`

- **egrep** is identical to **grep -E** and uses Extended Regular Expressions by default
- `$ egrep -i "[linux]{3,5}" -r /etc # identical to grep -i "[linux]\{3,5\}" and also with grep -E -i "[linux]{3,5}"`
- **fgrep** is identical to **grep -F** and matches the pattern literally irrespective of special characters
- `$ fgrep '[:alpha:]' regex.examples # Will search for the the string [:alpha:] instead of the alpha class. Identical to grep -F '[:alpha:]'`
- `$ fgrep "[linux]{3,5}" regex.examples # will search for the string [linux]{3,5} instead of the enclosed characters`

"You can't grep dead trees"
~ Ancient UNIX proverb ~



Filter and process text with `sed`

- **sed** is a powerful command for filtering and processing text
- Uses Basic Regular Expressions by default
- Uses Extended Regular Expressions when called with the **-r** option
- It has its own subcommands



Filter and process text with `sed`

- `$ sed -e 's/UUID/uuid/' /etc/fstab #`
replace the first occurrence of **UUID**, in a line, with **uuid**.
- `$ sed -e 's/UUID/uuid/g' /etc/fstab #`
replace all occurrences of **UUID** in a line, with **uuid**.
- `$ sed '1,4d' /etc/fstab > ~fstab.cribbled #`
remove lines 1 to 4 from **fstab**
- `$ sed '/^$/d' #` delete all empty lines
- `$ sed 'y/leti/1371/' matches.txt #` Replace characters **l,e,t** or **i** with **1371** respectively
- `$ sed -f rot13.sed matches.txt #` run **sed** commands from the **rot13.sed** file

Filter and process text with `sed`

Options:

- **-e** # defines a following command. It is optional when there is only one command
- **-f file#** reads command from **file** instead from the CLI
- **-r** # allows Extended Regular Expressions
- **-i, --inplace** # changes the file in place. Use with caution!

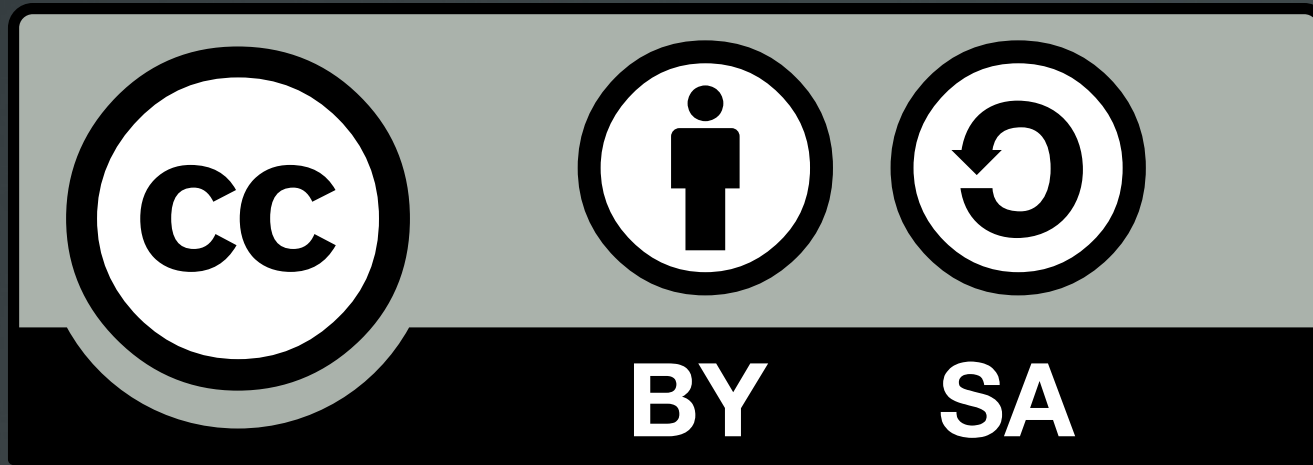


More info about regular expressions

- `$ man 7 regex`
- http://en.wikipedia.org/wiki/Regular_expression#Syntax
- <http://en.wikibooks.org/wiki/Regex>
- http://tldp.org/LDP/Bash-Beginners-Guide/html/chap_04.html



License



The work titled "LPIC-1 101-400 – Lesson 7" by Theodotos Andreou is distributed with the Creative Commons Attribution ShareAlike 4.0 International License.

