2.12.7 Content Search Facts

A Linux administrator must possess the skills to locate text within the contents of a file.

This lesson covers the following topics:

- File search commands
- The **diff** command

File Search Commands

The **grep** command searches through file text for specific words or character patterns. The following table describes the **grep**, **egrep**, and **fgrep** commands and several of their options.

Command	Description	Examples
grep	Searches through files for a specified character string. By default, **grep** is context sensitive and displays the string in the context of the line containing the string. • **-A** *[number]* prints a specified number of lines following the matching lines. • **-a** searches binary (executable) files as though they were text files. • **-B** *[number]* prints a specified number of lines before the matching lines. • **-C** *[number]* prints a specified number of lines of context around the matching lines. • **-c** shows the number of matches of the string for the file. • **-E** uses regular expressions for the text pattern. • **-e** *[pattern]* specifies a literal pattern. • **-f** searches for multiple strings using a file that lists the string patterns. • **-l** lists just the names of the files with a match. This is used to search multiple files. • **-m** *[number]* shows only a specified number of matches for a file.	• **grep -A 3 Midway ~/docs/WWII-report** searches WWII-report for the pattern "Midway" and prints the line and the next three lines. • **grep -a var11 /bin** searches all files, including binary files, in the /bin directory for the pattern "var11". • **grep -c Midway ~/docs/WWII-report** shows a number representing the number of times the pattern "Midway" was found in the WWII-report file. • **grep -C -3 Midway ~/docs/WWII-report** shows the specified number of lines preceding and following the matching lines. • **grep -e '--count' ~/docs/doc1** looks for the pattern "--count" in the doc1 file rather than interpreting it as an option. • **grep -l -r Midway ~/docs** shows the name of all files in the /home/user/docs directory that contain the term "Midway". • **grep -m 2 battle ~/docs/WWII-report** shows only the first two times the term "battle" is found in the file.

	• **-n** displays the line number of the lines containing the term. • **-r** searches the directory and all subdirectories for files containing the term. • **-v** displays non-matching lines. • **--include=*[file_name]*** searches only in files with names that match a specified string. • **--exclude=*[file_name]*** searches in files with names that do not match a specified string. • **-w** searches for whole words only.	• **grep -n -i customVariable1 ~/java/program1.java** shows the line numbers of lines that have the term "customVariable1" in the program1.java file. This ignores the case. • **grep -r battle ~/docs/** searches the directory and all subdirectories for the term "battle". • **grep -w tank ~/docs/WWII-report** searches only for the whole word "tank" in the file.
egrep	Uses regular expressions in the search strings. The **egrep** command uses the same options and syntax as **grep** and is identical to **grep -E**. Constructors for **egrep** regular expressions include: • **^** matches terms that occur at the beginning of a line. • **$** matches terms that occur at the end of a line. • **\<** matches words that begin with the term. • **\>** matches words that end with the term. • **[*asdf*]** matches any one of the characters in the brackets. • **[*0-9*]** matches any of the range of numbers 0-9. • **[^*xyz*]** omits any one of the letters in the list • **.** matches any single character. • **[*asdf*]+** matches one or more of the characters in the list. • ***** matches any number or none of the preceding single character. • **|** matches either of the terms. • **** displays the literal value of a character used for expressions. • **()** groups expressions.	• **egrep ^FAILURE ~/error_logs** matches the term "FAILURE" when it is at the beginning of the line in error_logs. • **egrep tty7$ ~/.bash_history** matches the term "tty7" when it is at the end of the line. • **egrep \<are ~/myfile** matches all words or strings that begin with "are" (e.g., "are", "area", and "arena"). • **egrep \>are ~/myfile** matches all words or strings that end with "are" (e.g., "are", "hare", and "aware"). • **egrep watche[ds] ~/myfile** matches either "watched" or "watches". • **egrep exhibit[0-9] ~/myfile** matches "exhibit1", "exhibit3", or "exhibit8". • **egrep [^Xx]mas ~/myfile** matches "Christmas" but not "xmas" or "Xmas". • **egrep .are ~/myfile** matches "hare" and "care", but not "aware" or "are". • **egrep file[0-9]+ ~/myfile** matches "file0", "file10", and "file15636". • **egrep fil* ~/myfile** matches "fil", "filll", and "fillllllllllllll". • **egrep fil.* ~/myfile** matches "file", "fill", "file102", and "filings".

		• **egrep men|women ~/myfile** matches "men" or "women". • **egrep Hello\? ~/myfile** matches "Hello?".
fgrep	Uses a file as the source for the string patterns. When searching for fixed strings rather than regular expressions, **fgrep**: • Uses the same options as the **grep** command and has the same syntax. • Is identical to **grep -F**, but searches faster than **grep**. • Interprets the pattern as a list of fixed strings, any of which can be matched.	• **fgrep Midway Nimitz ~/docs/myfile** searches myfile for lines containing "Midway" or "Nimitz".

The diff Command

The **diff** command is short for difference. It is used to display the differences between two files, line by line. The following table describes the **diff** command and several of its options:

Command	Description	Examples
diff	Displays the differences between two files, line by line. The output will contain the following characters: • < *[text]* : only the first file contains this text. • > *[text]* : only the second file contains this text. • a : text has been added. • c : text has changed. • d : text has been deleted. Options for the **diff** command include the following: • **-c** displays differences in context mode. • **-u** *[number]* prints a specified number of lines in a unified context. • **-i** ignores the case and treats uppercase and lowercase the same. • **-w** ignores all white space.	• **diff file1.txt file2.txt** displays the differences using the default display mode. • **diff -c file1.txt file2.txt** displays the differences using the context mode.

	• **-y** displays the output in two columns.	