**GREP COMMANNDS**

The grep command in Unix/Linux is a powerful tool used for searching and manipulating text patterns within files. Its name is derived from the ed (editor) command g/re/p (globally search for a regular expression and print matching lines), which reflects its core functionality. grep is widely used by programmers, system administrators, and users alike for its efficiency and versatility in handling text data.

**Syntax of grep Command**

The basic syntax of the `grep` command is as follows:

grep [**options**] pattern [**files**]

[**options**]: These are command-line flags that modify the behavior of grep.

**[pattern**]: This is the regular expression you want to search for.

[**file**]: This is the name of the file(s) you want to search within. You can specify multiple files for simultaneous searching**.**

**Options Available in grep Command**

| **Options** | **Description** |
| --- | --- |
| **-c** | This prints only a count of the lines that match a pattern |
| **-h** | Display the matched lines, but do not display the filenames. |
| **-i** | Ignores, case for matching |
| **-l** | Displays list of a filenames only. |
| **-n** | Display the matched lines and their line numbers. |
| **-v** | This prints out all the lines that do not matches the pattern |
| **-e exp** | Specifies expression with this option. Can use multiple times. |
| **-f file** | Takes patterns from file, one per line. |
| **-E** | Treats pattern as an extended regular expression (ERE) |
| **-w** | Match whole word |
| **-o** | Print only the matched parts of a matching line, with each such part on a separate output line. |
| **-A n** | Prints searched line and nlines after the result. |
| **-B n** | Prints searched line and n line before the result. |
| **-C n** | Prints searched line and n lines after before the result. |

**GREP Command Use Cases**

**01)** cat log-2023-06-28.0.log

This command displays the **entire contents** of the file log-2023-06-28.0.log in the terminal.  
It's typically used to **quickly view log files** or to pipe content into other commands for analysis.

**02)** grep "access" log-2023-06-28.0.log

This command searches the file log-2023-06-28.0.log for all lines that contain the word "access" and prints them to the terminal. It is useful for filtering specific log events, such as user access entries or API access attempts

**03)** grep "'dispatcherServlet'$" log-2023-06-28.0.log

This command searches the log-2023-06-28.0.log file for lines that end exactly with 'dispatcherServlet' (including the single quotes). The $ anchor ensures that the match is only for lines where 'dispatcherServlet' appears at the end of the line. Useful for identifying servlet completion logs or endpoints handled by the dispatcher.

**04)** grep -A1 access log-2023-06-28.0.logog

This command searches for lines in log-2023-06-28.0.log that contain the word access and also displays the line immediately after each match. -A1 means “After 1 line” — useful when you want to capture related log details that follow access events, such as status codes or response messages.

**05)** grep -i access log-2023-06-28.0.log

This command searches for the word access in the file log-2023-06-28.0.log, ignoring case sensitivity. The -i flag makes the search case-insensitive, so it will match access, Access, ACCESS, etc. Useful when log entries vary in case formatting but refer to the same event type.

**06)** grep -c access log-2023-06-28.0.log

This command counts the number of lines in log-2023-06-28.0.log that contain the word access. The -c flag stands for “count”. It returns a single number — useful for getting a quick summary of how many access-related events occurred in the log file**.**

**07)** grep "access" log-2023-06-28.0.log > temp.txt

This command searches for lines containing the word access in the file log-2023-06-28.0.log and writes the matching lines to a new file named temp.txt.

> redirects the output from the terminal into a file. If temp.txt already exists, it will be overwritten. Useful for saving filtered log data for reporting or further analysis.

**08)** cp -r files01 samplefiles01

This command copies the entire files01 directory and its contents recursively into a new directory called samplefiles01. cp = copy command -r = recursive (copies all subdirectories and files) If samplefiles01 doesn't exist, it will be created. Useful for duplicating folders for backup or testing purposes.

**09)** gunzip -k access.log.gz

This command decompresses the file access.log.gz and keeps the original .gz file intact. gunzip is used to extract .gz compressed files. -k stands for “keep”, so the original .gz file is not deleted after extraction. The result will be a new file named access.log in the same directory. Useful when you want to inspect the contents but preserve the compressed archive.

**10)** gzip -d access.log.gz

This command decompresses the file access.log.gz using the gzip utility and removes the original .gz file after extraction. -d stands for “decompress”. The output file will be access.log. Use this when you want to extract the contents and don’t need to keep the compressed version.

**11)** zcat access.log.gz

used to decompress and display the contents of the compressed file access.log.gz to the standard output (your terminal).