The grep command is used for searching text and patterns in files and command output. It supports regular expressions, case-insensitive search, recursive search, and more.

Table of Contents

Basic grep Usage

Command	Description
grep “word” file.txt	Search for “word” in file.txt.
grep -i “word” file.txt	Case-insensitive search.
grep -w “word” file.txt	Match whole word only.
grep -n “word” file.txt	Show line numbers with matches.
grep -c “word” file.txt	Count occurrences of “word”.
grep -v “word” file.txt	Show lines that do NOT contain “word”.

Example: Case-insensitive search for “error” in a log file

```
grep -i "error" /var/log/syslog
```

Searching in Multiple Files

Command	Description
grep “word” file1.txt file2.txt	Search in multiple files.

Command	Description
grep -l “word” *.txt	Show only filenames containing “word”.
grep -r “word” /path/to/dir	Recursively search inside all files in a directory.
grep -Rl “word” /path/to/dir	List filenames with matches (no content).

Example: Recursively find “TODO” comments in code

```
grep -r "TODO" ~/projects/
```

Using Regular Expressions

Command	Description
grep “^word” file.txt	Find lines starting with “word”.
grep “word$” file.txt	Find lines ending with “word”.
grep “[0-9]” file.txt	Find lines containing numbers.
grep “error.*warning” file.txt	Match “error” followed by “warning”.
`grep -E “error	fail” file.txt`

Example: Find lines that contain “log” at the beginning

```
grep "^log" logfile.txt
```

Filtering Command Output

Command	Description
`ps aux	grep “firefox”`
`ls -l	grep “^d”`
`df -h	grep “/dev/sd”`
`cat file.txt	grep -v “^#”`

Example: Find all processes related to “apache”

```
ps aux | grep "apache"
```

Grep with Context (Surrounding Lines)

Command	Description
grep -A 3 “error” file.txt	Show 3 lines after each match.
grep -B 3 “error” file.txt	Show 3 lines before each match.
grep -C 3 “error” file.txt	Show 3 lines before & after match.

Example: Show errors with 2 lines of surrounding context

```
grep -C 2 "error" logfile.txt
```

Advanced Grep Techniques

Command	Description
grep -o “[0-9]\+” file.txt	Extract only numbers from a file.
`grep -E “ERROR	WARN” logfile.txt`
grep –color=auto “pattern” file.txt	Highlight matched text.
grep -P “\d{3}-\d{2}-\d{4}” file.txt	Find social security numbers (-P for Perl regex).

Example: Extract email addresses from a file

```
grep -E "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}" emails.txt
```

Saving and Redirecting Grep Output

Command	Description
grep “error” logfile.txt > errors.txt	Save results to a file.
grep “error” logfile.txt >> errors.txt	Append results to a file.
`grep “error” logfile.txt	tee errors.txt`

Example: Save all warning messages to “warnings.log”

```
grep "WARNING" /var/log/syslog > warnings.log
```

Summary

Feature	Command Example
Basic Search	grep “word” file.txt
Case Insensitive	grep -i “word” file.txt
Recursive Search	grep -r “word” /path/to/dir
Show Line Numbers	grep -n “word” file.txt
Search Multiple Patterns	`grep -E “error
Show Before/After Context	grep -C 3 “error” file.txt
Extract Numbers	grep -o “[0-9]\+” file.txt

This Linux grep Cheat Sheet will help you find text efficiently in files, logs, and command output.