UBUNTU REGEX BASH

EGREP LIKE A LINUX PRO

Extended regex · access.log · error.log · pattern groups

```
user@ubuntu: ~/logs

$ egrep -E   "POST|PUT|DELETE"    access.log
$ egrep -i   "error|fail|denied"     error.log
$ egrep -o   "[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+"access.log
$ egrep -c   "^(GET|HEAD)"    access.log
$ _
```

30:00

UBUNTU REGEX BASH

EGREP LIKE A LINUX PRO

Extended regular expressions for fast, precise log filtering

presenter's terminal

```
$ ready to teach students real-world text processing
```

UBUNTU REGEX BASH

EGREP LIKE A LINUX PRO

Extended regular expressions for fast, precise log filtering

presenter's terminal

```
$ ready to teach students real-world text processing
```

What students will learn

1. **Why egrep**

   extended regex vs basic grep

2. **Syntax basics**

   egrep pattern file structure

3. **Core options**

   -E -i -v -c -o -w -l -n -r

4. **Extended metacharacters**

   | + ? ( ) { }

5. **Real log examples**

   Apache access.log & error.log

6. **Cheat sheet**

   everything on one slide

Why teach egrep instead of plain grep

1

No escaping

Extended regex lets you use |, +, ?, (), and {} directly — no backslash soup like grep -E needs to fake.

2

Multi-pattern search

Match several patterns in one pass: POST|PUT|DELETE finds three HTTP methods with a single command.

3

Same as grep -E

egrep is just grep -E under the hood — students learn one mental model that also explains modern grep usage.

Syntax breakdown

```
egrep [OPTIONS] 'PATTERN' file
```

`egrep`	invokes extended regex matching (equivalent to grep -E)
`[OPTIONS]`	flags that change matching behavior or output format
`'PATTERN'`	an extended regular expression, quoted to protect it from the shell
`file`	the file(s) to search — omit to read from standard input

Core options to demonstrate

Option	Meaning
`-E`	Use extended regex (default for egrep, explicit for grep -E)
`-i`	Case-insensitive match
`-v`	Invert match — show lines that do NOT match
`-c`	Print only a count of matching lines
`-o`	Print only the matched text, not the whole line
`-w`	Match whole words only
`-n`	Show line numbers with matches
`-r / -R`	Recursively search a directory tree
`-l`	Print only filenames that contain a match
`--color`	Highlight the matched text in the terminal

EXAMPLE 1

Find multiple HTTP methods in one pass

Extended regex lets you OR patterns together without escaping the pipe.

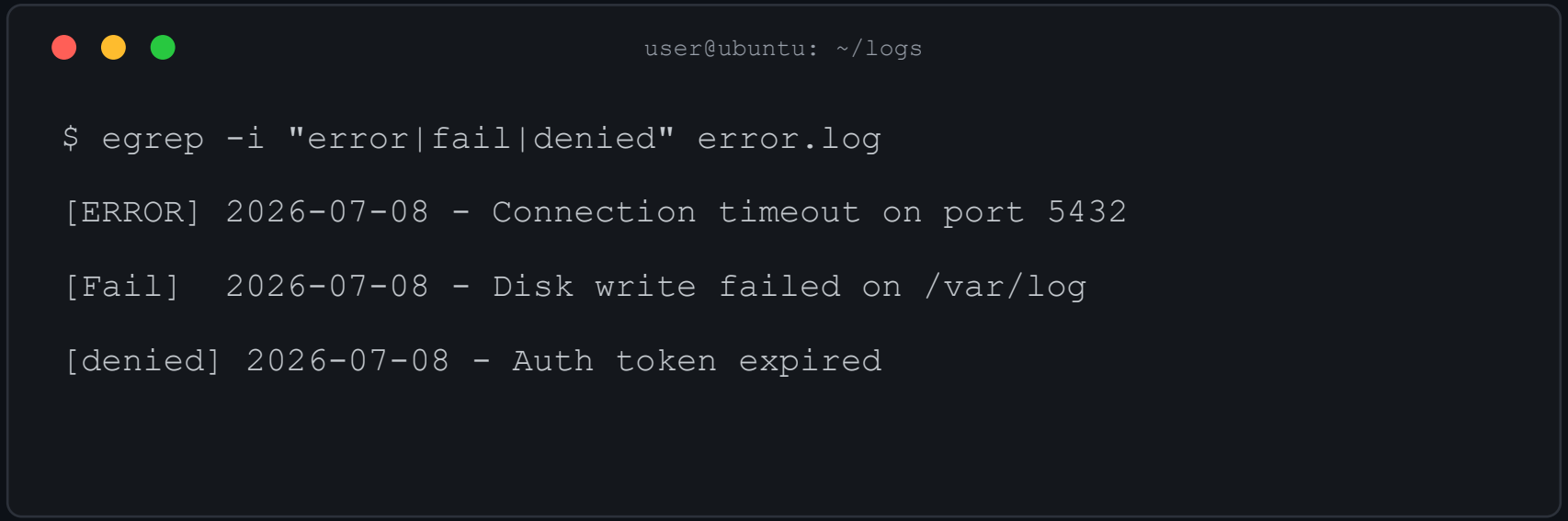
user@ubuntu: ~/logs

```
$ egrep -E "POST|PUT|DELETE" access.log
192.168.1.4 - - [08/Jul/2026] "POST /login HTTP/1.1" 200
192.168.1.9 - - [08/Jul/2026] "DELETE /cart/2 HTTP/1.1" 204
192.168.1.2 - - [08/Jul/2026] "PUT /profile HTTP/1.1" 200
```

EXAMPLE 2

Filter error severity levels

Case-insensitive extended matching catches ERROR, error, and Error in one command.



EXAMPLE 3

Extract every IP address from a log

Combine -o (print only the match) with a grouped regex to pull structured data out of raw text.

user@ubuntu: ~/logs

```
$ egrep -o "[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+" access.log
192.168.1.4
192.168.1.9
192.168.1.2
```

egrep cheat sheet

`egrep -E`	extended regex mode
`egrep -i`	ignore case
`egrep -v`	invert match
`egrep -c`	count matches
`egrep -o`	print match only
`egrep -w`	whole word match
`egrep -n`	show line numbers
`egrep -r`	search recursively
`| + ? ( ) {}`	extended metacharacters
`\1 \2`	backreferences to groups

You now know egrep

- Extended regex removes the need to escape |, +, ?, (), and {}
- -E, -i, -v, -c, -o, and -w cover most real-world filtering tasks
- Practice on your own access.log and error.log files with the commands above

Next in the series: sed →