Name:-Vaibhav kumar gupta

Date:-20-02-2025

Linux:-

Regular Expressions (RegEx) are patterns used to match, search, and manipulate text in Linux. They are commonly used with commands like grep, sed, awk, and perl.

Types of Regular Expressions in Linux

There are **three types** of regular expressions in Linux:

1. **Basic Regular Expressions (BRE)** → Used with grep, sed, awk
2. **Extended Regular Expressions (ERE)** → Used with grep -E, sed -r, awk
3. **Perl-Compatible Regular Expressions (PCRE)** → Used with grep -P, perl

Basic RegEx Syntax & Examples

Here are key RegEx metacharacters and their examples:

Matching Literal Characters

A simple match of a word or character.

echo "hello world" | grep "hello"

Output:

hello world

Finds the exact word "hello".

Anchors (^ and $)

^ → Matches the beginning of a line

$ → Matches the end of a line

echo -e "hello\nworld" | grep "^hello"

Output:

Hello

"hello" appears at the start of a line.

echo -e "hello\nworld" | grep "world$"

Output:

world

"world" appears at the end of a line.

Character Classes ([])

Matches any one character inside the brackets.

Example: [aeiou] → Matches any vowel.

echo "hello" | grep "[aeiou]"

Output:

hello

Matches "e", "o" in "hello".

Negating a Character Class ([^])

[^aeiou] → Matches any character except vowels.

echo "hello" | grep "[^aeiou]"

Output:

hll

Matches only consonants (h, l, l).

Wildcard (.)

.Matches any single character except a newline (\n).

echo "cat bat hat" | grep "c.t"

Output:

cat

Matches c.t, where . can be any character.

Quantifiers (*, +, ?, {})

Quantifiers define how many times a character or group appears.

Symbol	Meaning	Example
*	Matches 0 or more times	go* → g, go, goo, gooo
+	Matches 1 or more times	go+ → go, goo, gooo
?	Matches 0 or 1 times	colou?r → color, colour
{n}	Matches exactly n times	o{2} → oo
{n,}	Matches n or more times	o{2,} → oo, ooo, oooo
{n,m}	Matches between n and m times	o{2,4} → oo, ooo, oooo

Example 1: * (0 or more)

bash

```
echo "go goo gooo" | grep "go*"
```

Output:

```
go
```

go goo gooo

"go*" matches "g", "go", "goo", "gooo".

Example 2: + (1 or more)

echo "go goo gooo" | grep -E "go+"

Output:

goo

gooo

"go+" matches "goo", "gooo" (but not "go" alone).

Example 3: {} (Exact Match)

bash

echo "aaa aa aaaa" | grep -E "a{3}"

Output:

aaa

aaaa

"a{3}" matches "aaa" and "aaaa" (because "a{3}" means 3 or more).

Grouping with Parentheses ()

Used to group patterns together.

Example: Finding "go" repeated twice.

bash

echo "gogogo gogo" | grep -E "(go){2}"

Output:

gogo

gogogo

Matches "gogo" and "gogogo".

Alternation (|)

| (OR operator) matches either pattern.

echo "apple banana orange" | grep -E "apple|orange"

Output:

apple orange

Matches "apple" or "orange".

Word Boundaries (\b)

\b matches the start/end of a word.

echo "hello hell" | grep -E "\bhello\b"

Output:

hello

Matches "hello" but not "hell".

Escape Special Characters (\)

If you need to match special characters, use \ (backslash) to escape them.

echo "1+1=2" | grep "1\+1"

Output:

1+1=2

Escapes +, so it matches "1+1".

🔷 Using Regular Expressions in Commands

🔶 grep

Basic pattern search

bash

grep "word" file.txt

Case-insensitive search

grep -i "word" file.txt

Match only whole words

grep -w "word" file.txt

Extended regex (-E)

grep -E "go+|run" file.txt

🔶 sed (Stream or)

Replace "hello" with "hi"

sed 's/hello/hi/g' file.txt

Delete lines matching "error"

sed '/error/d' file.txt

🔶 awk

Print lines containing "error"

awk '/error/ {print}' file.txt

Print only the first word of each line

awk '{print $1}' file.txt

🔷 Summary Table

Symbol	Description	Example
.	Any single character	a.b → acb, a3b
^	Start of line	^Hello
$	End of line	world$
*	0 or more occurrences	go*
+	1 or more occurrences	go+

? 0 or 1 occurrence colou?r

{n,m}Between n and m occurrences o{2,4}

` ` OR operator

\b Word boundary \bgo\b

\ Escape special char \.