Session - 7

grep in Linux — Search Command Explained

🔍 What is grep?

grep (Global Regular Expression Print) is a powerful **command-line tool** used in Linux/Unix to **search for patterns** (like text or characters) **within files or output**.

🛠️ Basic Syntax

bash

```
grep [OPTIONS] PATTERN [FILE...]
```

- PATTERN: The string or regular expression to search for
- FILE: One or more files to search in

📌 Examples

🔹 1. Search for a word in a file

bash

```
grep "apple" fruits.txt
```

Finds and prints lines containing **"apple"** in fruits.txt.

- **2. Search in multiple files**

bash

grep "main" *.c

Finds lines with **"main"** in all .c files.

- **3. Ignore case (-i)**

bash

grep -i "linux" notes.txt

Matches **Linux**, **LINUX**, **linux**, etc.

- **4. Show line numbers (-n)**

bash

grep -n "hello" script.sh

Displays matching lines with their **line numbers**.

- **5. Count matches (-c)**

bash

grep -c "error" logfile.txt

Shows the **number of lines** containing "error".

- **6. Display lines that do not match (-v)**

bash

grep -v "pass" students.txt

Shows lines **not** containing "pass".

🔹 7. Use Regular Expressions

bash

grep "^[A-Z]" file.txt

Matches lines that **start with a capital letter**.

🔹 8. Recursive search in all files in a directory (-r)

bash

grep -r "main()" .

Recursively finds "main()" in all files under current directory.

🔧 Commonly Used Options

Option	Meaning
-i	Ignore case
-v	Invert match (show non-matching lines)
-n	Show line numbers
-c	Show only count of matching lines
-r	Recursive search in directories
-l	Show file names with matches

Option	Meaning
-w	Match **whole words** only
-e	Use multiple patterns

✅ **Practical Use Case Example**

bash

CopyEdit

dmesg | grep -i "usb"

Filters USB-related system logs.