

# Devops-II



# BASH SCRIPTING.



Bourne Again Shell - It is a command processor where user commands causes actions.

## PURPOSE OF BASH

- \* **Automate** different types of task in a system.
  - Developers can avoid repetitive tasks by using BASH.
  - Supports Variables, Conditional statements and loops like any programming language.

## KEY - FEATURES (ISVCF)

- \* **Interactivity**: Users interact with system using CLI.
- \* **Scripting**: Users can write Scripts to Automate tasks
- \* **Variables**: Store & Manipulate data using variables.
- \* **Control Structure**: Conditional statements and loops.
- \* **Function**: Reusability.

### Advantages

- It is simple.
- It helps to avoid doing repetitive tasks
- Easy to use
- Frequently performed tasks can be automated
- A sequence of commands can be run as a single command.

### Disadvantages

- Any mistake while writing can be costly.
- A new process launched for almost every shell command executed
- Slow execution speed
- Compatibility problems between different platforms

## 10 - General purpose Bash commands

- #1 - cd → change Directory
- #2 - mv → move/rename - file
- #3 - rm → Remove / Delete - file
- #4 - mkdir
- #5 - rmdir
- #6 - cat → Display Content of a file
- #7 - echo → Display msg
- #8 - grep → Search for pattern in file
- #9 - clear → Clean up terminal
- #10 - history → Prev commands used in terminal

# SPECIAL VARIABLES

## #1 - Positional Parameters: \$0, \$1, \$2

Eg: echo "Script-name: \$0"  
echo "First-arg: \$1"

## #2 - Environment Variables: \$HOME, \$USER, \$PATH

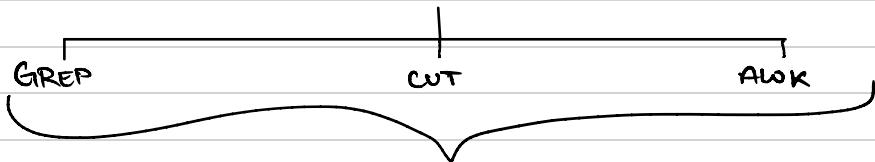
Eg: echo "Home Directory: \$HOME"

## #3 - Arithmetic Operations:

Eg: num1=10  
num2=5  
sum=\$((num1+num2))  
echo "Sum: \$sum"

## #4 - Local and Global Scope

# TEXT PROCESSING COMMANDS



Essential tools for manipulating & Extracting Information from Text data.

## GREP (Global Regular Expression Print)

- \* Searches for patterns in text-files and returns matching lines
- \* Syntax: GREP [option] pattern [filename]  
↓                  ↓  
grep "Keyword" filename

## CUT :

- \* Removes specific section from each line of a file.

- \* Syntax: cut [Options] [file]

cut -d ':' -f1,3 filename.csv  
↓                  ↓  
delimiter      Extract  
                  1st  
                  2nd  
                  3rd field  
name,age,city,country  
                  name,city

## AWK :

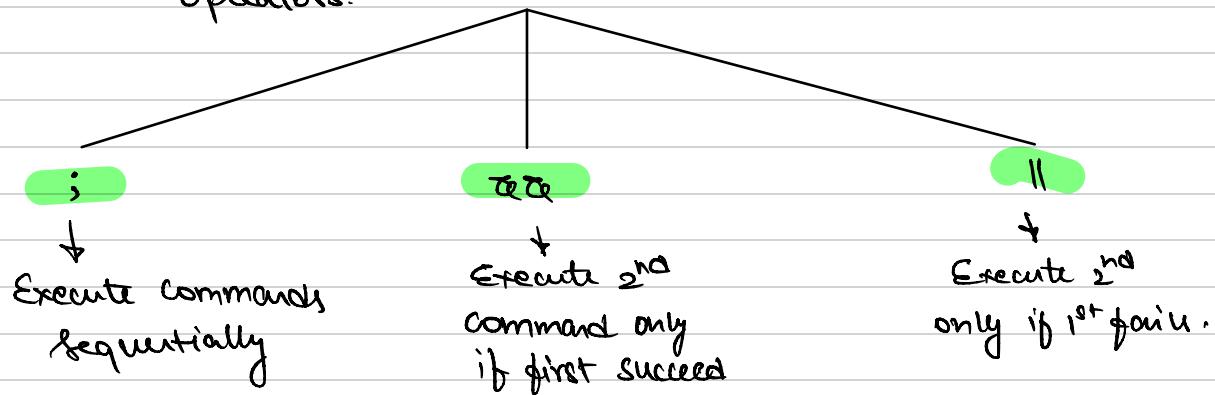
- \* Versatile text-processing tool for text-Scanning and processing

# INPUT & OUTPUT REDIRECTION

> - Output  
< - Input  
++ - Append

# Command Chaining

- ↳ Involves executing multiple commands sequentially or conditionally
- ↳ Operators:



# CRONTAB → Used for Schedule job.

Syntax: minute hour day month day commands  
of week