LINUX

QA for PEP Classes

Prepared by

Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

System Programming

Lovely Professional University

What is a shell?

In computing, a shell is a user interface that allows users to interact with an operating system's services and functions. It serves as an intermediary between the user and the kernel (the core part of the operating system). A shell takes commands from the user via a command-line interface (CLI) or a graphical user interface (GUI), interprets them, and then communicates with the operating system to execute those commands.

- Shells come in various types, such as:
  - Command-Line Shells: These allow users to interact with the operating system by typing text-based commands. Examples include Bash (Bourne Again SHell), PowerShell, and Zsh (Z Shell).
  - Graphical Shells: These provide a more visual interface, allowing users to interact with the operating system through icons, menus, and windows. Examples include Windows Explorer (on Microsoft Windows) and GNOME Shell (on Linux systems using the GNOME desktop environment).

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

How is the Linux file system structured?

The Linux file system is structured in a hierarchical manner, with directories (folders) organized in a tree-like structure. The root directory, denoted by "/", is at the top of this structure and serves as the starting point for the entire file system. Here is an overview of the main directories and their purposes:

- / (Root Directory): The top-level directory in the Linux file system. All other directories and files are subdirectories or files within the root directory.
- /bin (Binary): Contains essential system binaries (executable files) required for basic system functioning. Common commands like ls (list directory contents), cp (copy), and mv (move) reside here.
- /boot: Contains boot loader files and the Linux kernel, which is necessary to boot the operating system.
- /dev (Devices): Contains device files that represent hardware devices connected to the system, such as hard drives, USB devices, terminals, etc.
- /etc (Etcetera): Contains system-wide configuration files for various programs, services, and system settings.
- /home: Home directories for individual users are located here. Each user typically has a subdirectory within /home with their username, where they store personal files and configurations.
- /lib (Library): Contains system libraries—shared code used by programs at runtime.
- /media: Mount point for removable media like USB drives, DVDs, etc.
- /mnt (Mount): A directory used for temporarily mounting file systems or devices.
- /opt (Optional): Typically used for installing optional software or additional packages.
- /proc: A virtual file system that provides information about system processes and kernel configuration.
- /sbin (System Binaries): Contains system binaries essential for system administration tasks. Commands here often require administrative privileges.
- /tmp (Temporary): Used for temporary files created by programs or system processes. Files here are typically deleted upon system reboot.
- /usr (User): Contains user-related programs, libraries, documentation, etc. Often includes subdirectories like /usr/bin for user binaries, /usr/lib for user libraries, and /usr/share for shared data.
- /var (Variable): Contains variable data that frequently changes during system operation, such as log files, spool directories (for print queues), databases, etc.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What purposes does the history command serve?

The history command in a terminal or command-line interface displays a list of previously executed commands by the current user. Its primary functions include:

- Viewing Command History: It shows a chronological list of previously typed commands in the terminal session.
- Repeating or Recalling Commands: Users can easily rerun specific commands from their history by referencing their numerical order in the list.
- Efficient Command Recall: It allows users to search for specific commands or filter the command history based on keywords or patterns, making it easier to find and reuse previously used commands.
- Customization and Configuration: Users can adjust the behavior of the history command by setting options like the maximum number of commands to retain, controlling timestamps, ignoring duplicates, etc., via configuration files.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is initialization?

In Linux, initialization refers to the process of starting the operating system. It involves a series of steps that the system goes through when booting up to initialize and configure various components to prepare the system for user interaction. Initialization encompasses several stages:

- Boot Loader: The boot loader, such as GRUB (Grand Unified Bootloader), is the initial program loaded by the computer upon powering on. It helps load the Linux kernel into memory.
- Kernel Initialization: Once loaded, the Linux kernel initializes essential hardware components like CPU, memory, devices, and drivers required for the operating system to function.
- Init Process: After the kernel initializes, the init process, traditionally managed by programs like SysV init or modern replacements like systemd, takes control. It's responsible for starting system services, launching daemons, and setting up the user environment.
- Runlevels or Targets: The init process configures the system to a specific runlevel or target, determining which services and processes are started or stopped. Different runlevels or targets can define the system's behavior, such as single-user mode, multi-user mode with or without a graphical interface, or emergency mode.
- Service Initialization: During boot, the init process starts various system services and daemons according to the runlevel or target, ensuring that essential programs and services are running to support user applications.
- User Environment Setup: Finally, once the system is initialized and services are running, the user environment is prepared, allowing users to log in and interact with the system through a graphical interface or command-line interface.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What does 'su' stand for in Linux?

- In Linux, 'su' stands for "substitute user" or "switch user." The 'su' command allows a user to change to another user account, either temporarily or permanently, by providing the credentials (such as password) of the target user. By default, if no username is specified, 'su' switches to the superuser or root account, granting extensive system-level privileges. This command is commonly used to perform administrative tasks or access permissions restricted to other users or the root user.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Explain the significance of the 'su' command in multi-user environments.

- In multi-user environments, the 'su' command allows users to temporarily switch to other accounts, particularly the root user, enabling them to perform administrative tasks, collaborate, and manage user privileges while enhancing system security by limiting access to privileged operations.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Explain the differences between 'su -' and 'su'.

- 'su -' provides a full login environment, while 'su' alone switches users without altering the environment settings or directory.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Differentiate between 'su' and 'sudo' commands.

- 'su' switches to another user account after authentication, whereas 'sudo' grants temporary elevated privileges to execute specific commands without needing the target user's password, providing a more controlled approach to access system privileges.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Can you restrict certain users from using the 'su' command? If so, how?

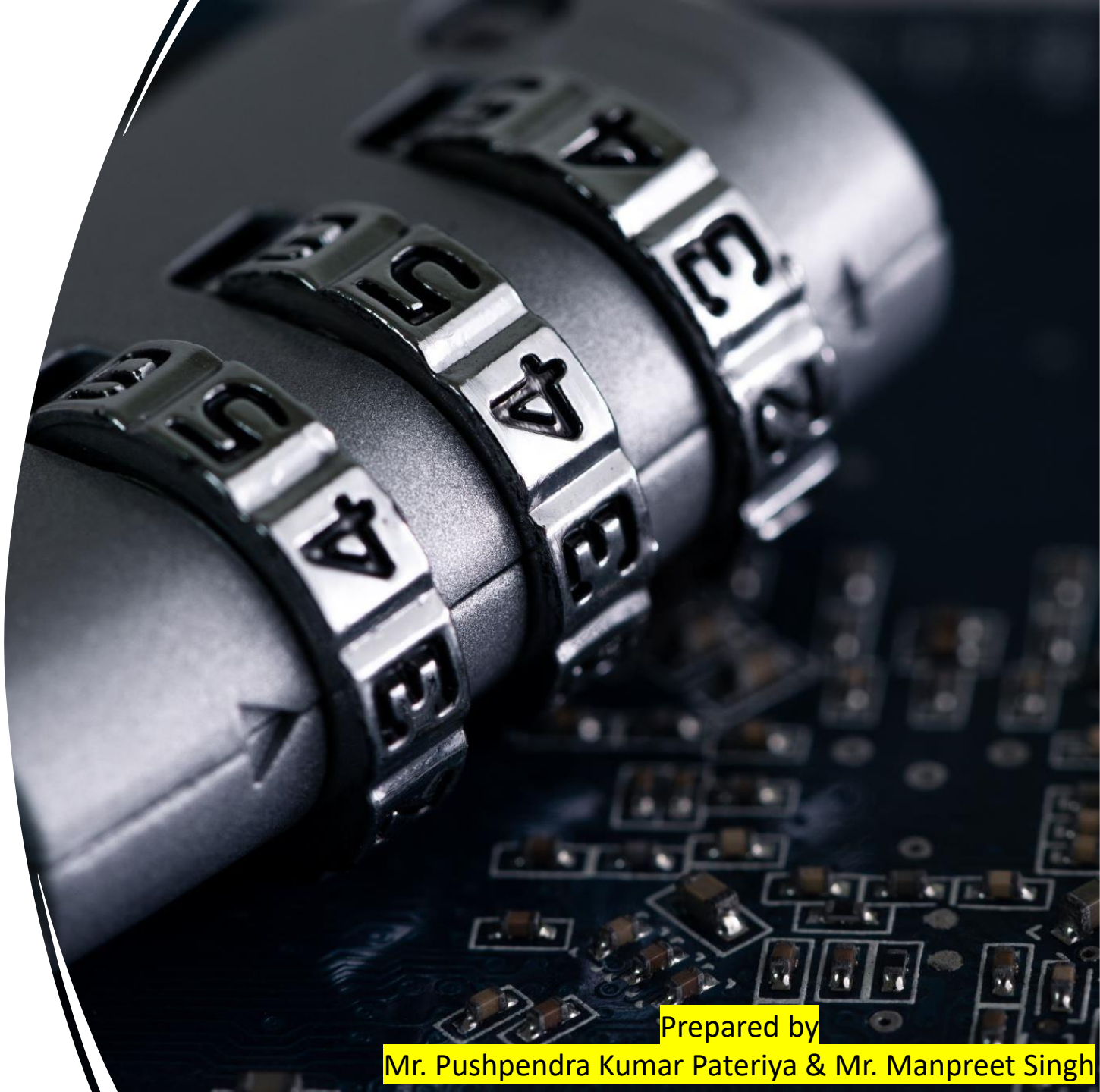
Yes, you can restrict certain users from using the 'su' command by modifying the file permissions for the 'su' executable or by managing user privileges using the '/etc/sudoers' file.

1. Modify file permissions for 'su':
   - sudo chmod o-x /bin/su
2. Manage user privileges using sudoers file:
   - sudo visudo
   - restricteduser ALL = /bin/su, !/bin/su

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

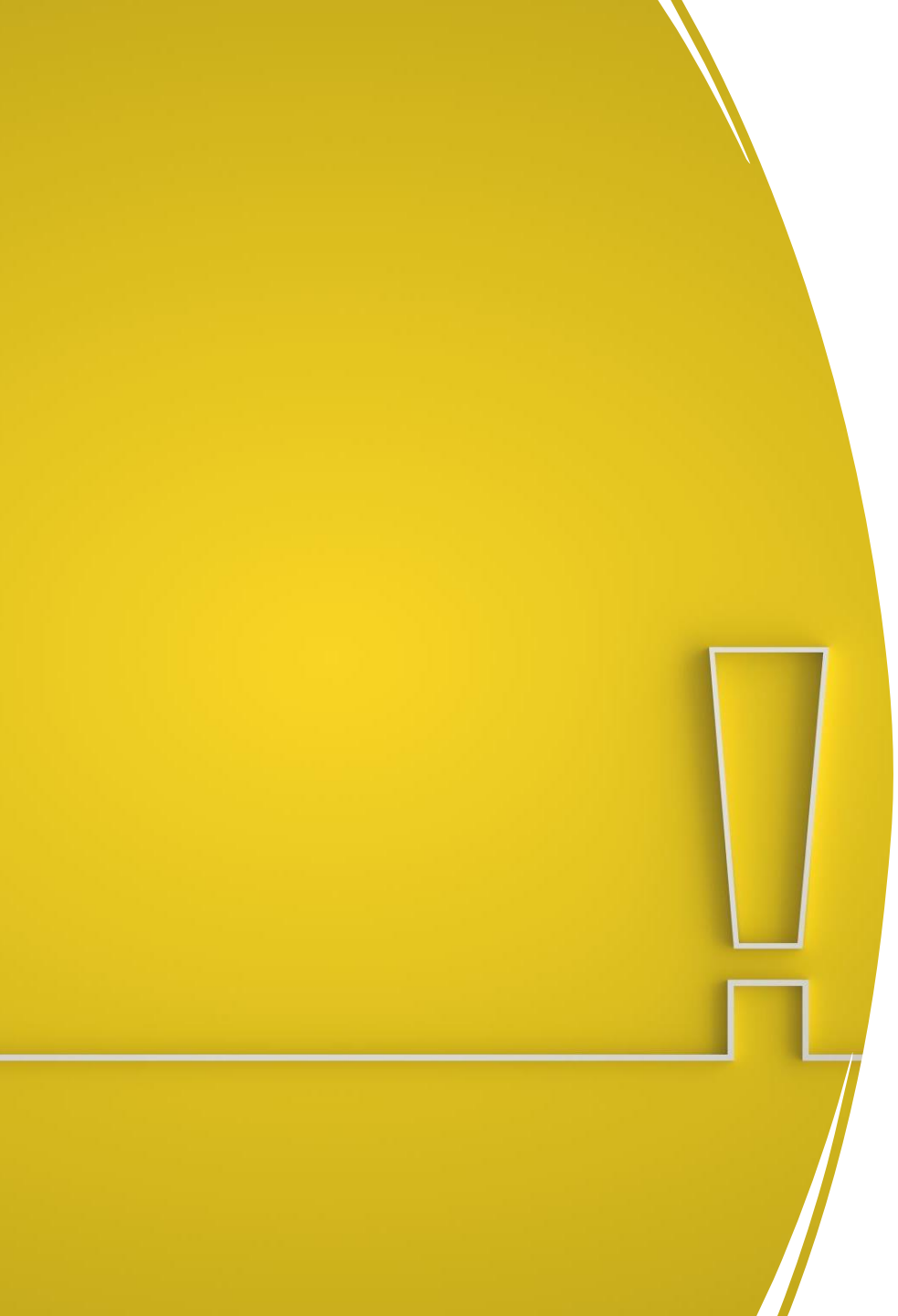
How can you exit from the 'su' session and return to the original user's environment?

To exit from an 'su' session and return to the original user's environment in Linux, you can use the 'exit' command or press 'Ctrl + D'.



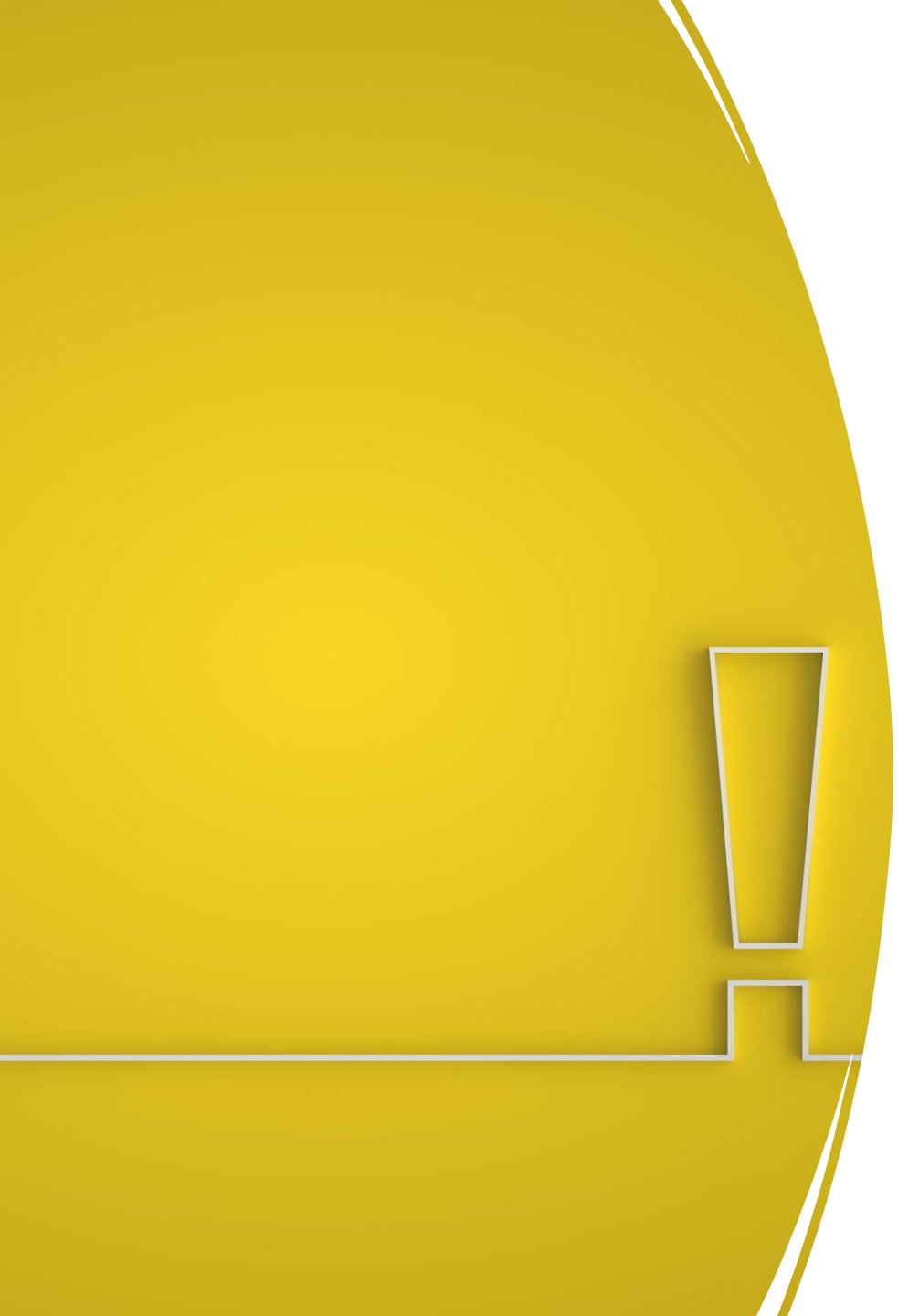
Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What does the 'head' command do in Linux?



- The 'head' command in Linux is used to display the beginning or the first few lines of a text file or input received from another command.
- By default, it shows the first 10 lines of a file, but this behavior can be altered using command options.
- 'head' is particularly useful when you need to quickly preview the content at the beginning of a file without displaying the entire file content.

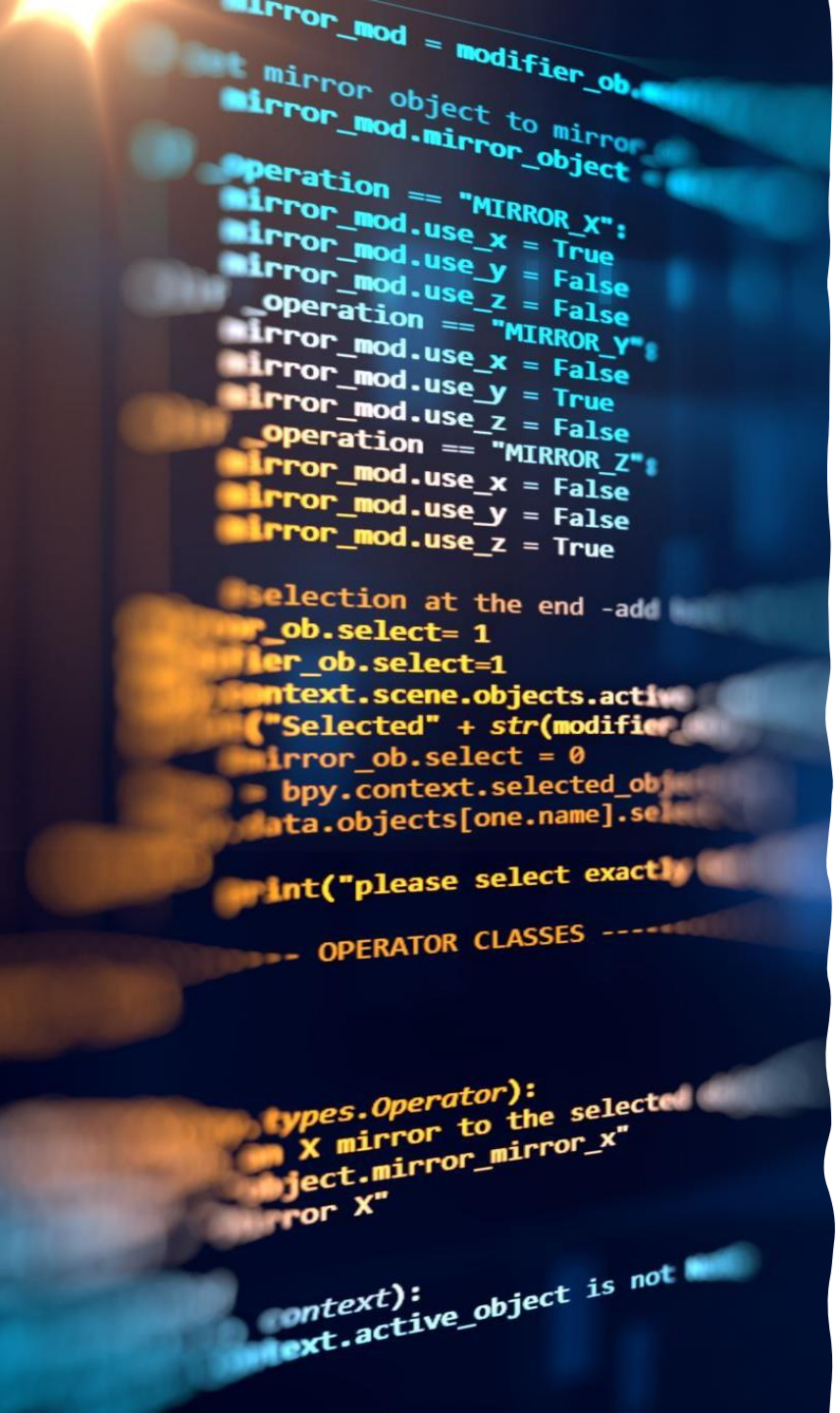
Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh



What happens if you use the '-n' option with 'head'? Provide an example.

- The '-n' option with the 'head' command in Linux allows you to specify the number of lines to display from the beginning of a file or input. It modifies the default behavior of displaying the first 10 lines.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh



What is the primary function of the 'tail' command in Linux?

- The primary function of the 'tail' command in Linux is to display the end or the last few lines of a text file or input received from another command.
- By default, it shows the last 10 lines of a file, but similar to 'head', this behavior can be modified using command options.
- 'tail' is particularly useful when you need to check the most recent entries in log files, monitor changes in real-time, or quickly view the end portion of a file without displaying the entire content.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What happens when you use the '-f' option with 'tail'? Provide an example.

- The '-f' option with the 'tail' command in Linux stands for "follow". When used, it allows 'tail' to continuously monitor a file for any new lines appended to it and display those new lines in real-time as they are added to the file. This option is commonly used to monitor log files or track ongoing changes in files.
- For example, to continuously monitor a log file named 'logfile.log' for any updates or additions to the file, you would use the '-f' option as follows: tail -f logfile.log
- When this command is executed, 'tail' starts displaying the content of 'logfile.log' initially. Then, as new lines are added to 'logfile.log', 'tail' will automatically update and display those new lines in real-time without exiting the command. This allows you to continuously monitor the file for changes as they occur. To exit this mode, you can typically press 'Ctrl + C' in the terminal.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

How do I get the list of processes currently running?

MacOS / Linux / Unix:

1. Terminal:
   - Open Terminal.
   - Use the ps command:
     - ps aux or ps -ef for a detailed list of processes with their information.
2. htop (Linux/Unix):
   - Install if not available: sudo apt-get install htop (for Debian-based systems).
   - Run htop command in Terminal for an interactive and detailed view of running processes.

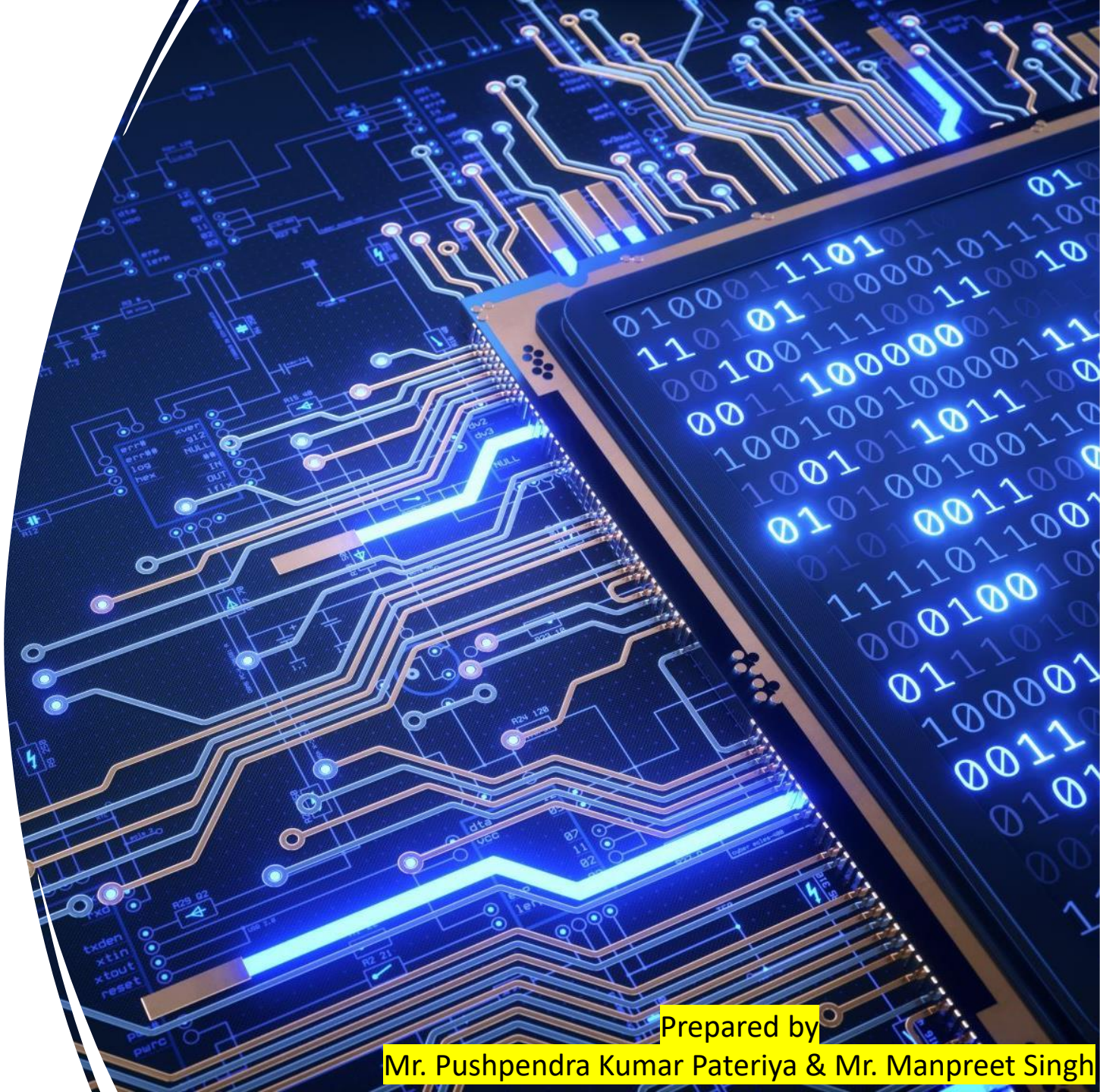
Windows:

1. Task Manager:
   - Press Ctrl + Shift + Esc to open Task Manager.
   - Go to the "Processes" tab to view the list of running processes.
2. Command Prompt:
   - Open Command Prompt (cmd.exe).
   - Type tasklist and press Enter to display a list of running processes.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

How can you filter the output of the ps command to display processes for a specific user?

Answer: To filter the output by a specific user, the -u option can be used with the ps command. For instance, ps -u username will display processes associated with the specified username.



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

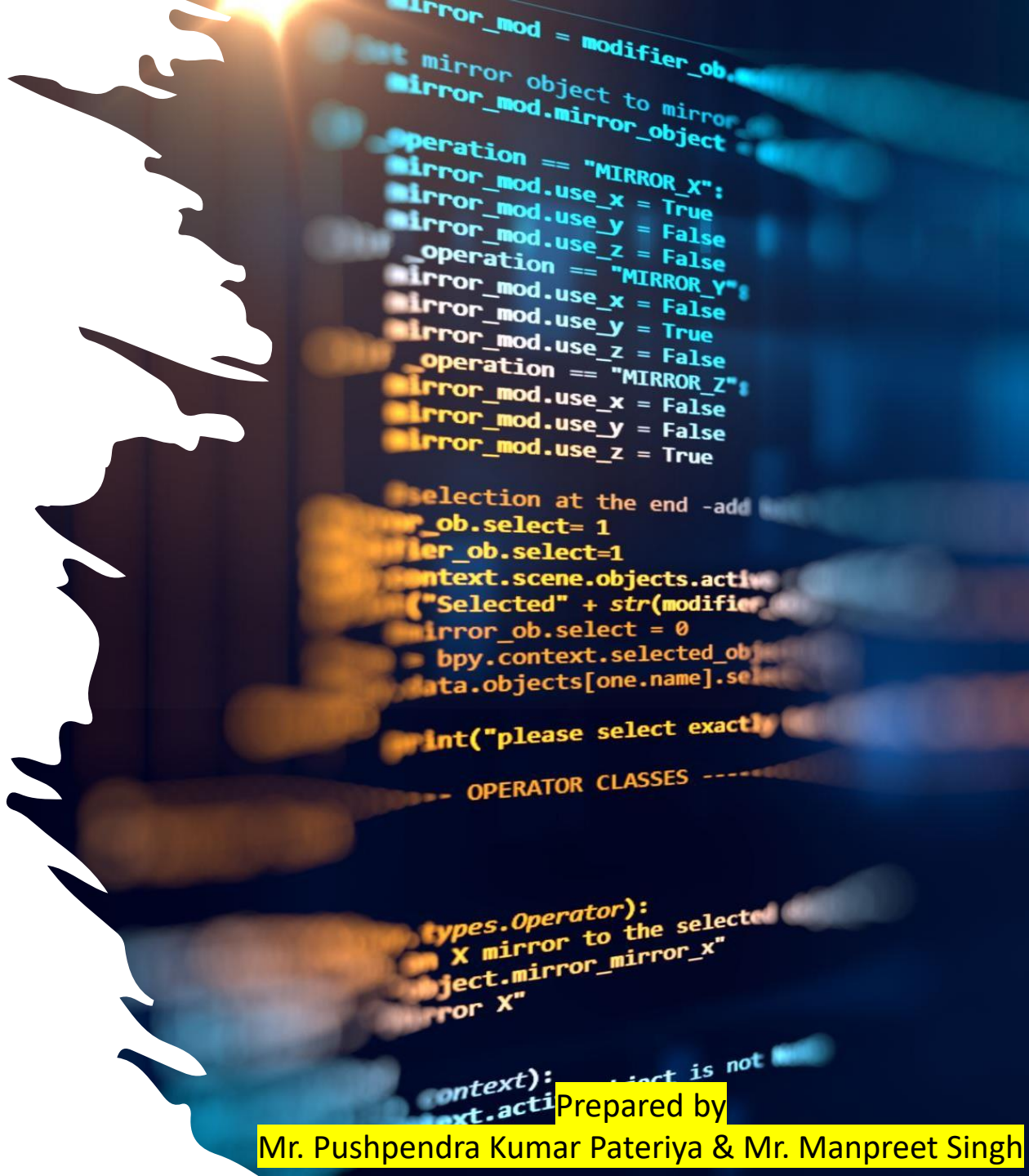
Explain how to display a hierarchical view of processes using the ps command.

Answer: The hierarchical view of processes can be displayed using the ps axjf command. It represents the relationship between parent and child processes in a tree-like structure, showing their PIDs and relationships.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Explain how you might limit the ps output to a specific process ID (PID).

Answer: To display information about a specific process ID, the -p option followed by the PID can be used. For instance, ps -p 1234 will display details about the process with PID 1234.



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

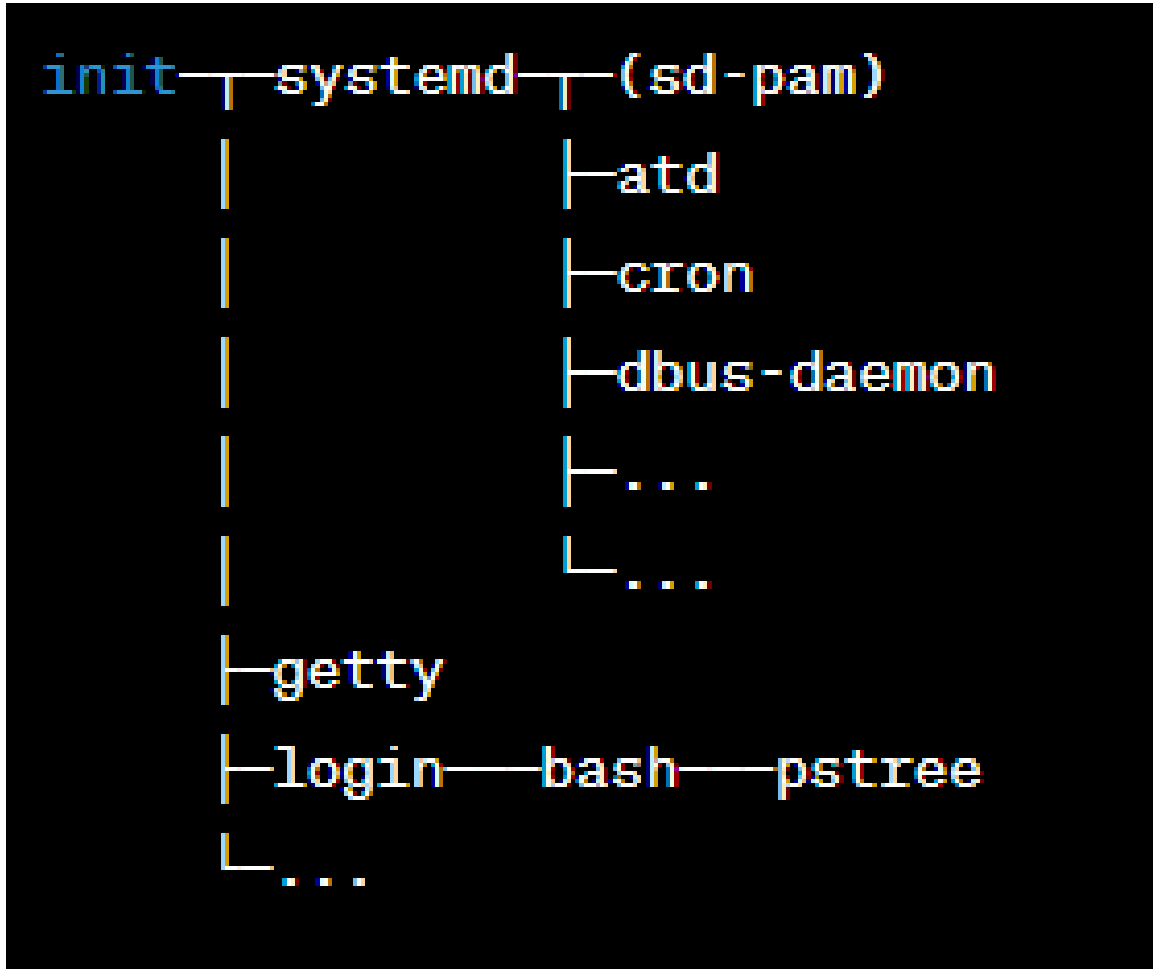
What is the difference between ps and top commands?

Answer: ps and top both display information about processes, but top provides a dynamic, real-time view of processes and system usage, continuously updating its display. In contrast, ps provides a snapshot of processes at the time it's executed.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is a pstree? How does it work?

- pstree is a command used in Unix-like operating systems (such as Linux) to display the processes in a tree-like structure, showing the hierarchical relationship between parent and child processes.
- Here's an example of how pstree might display the process hierarchy:



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

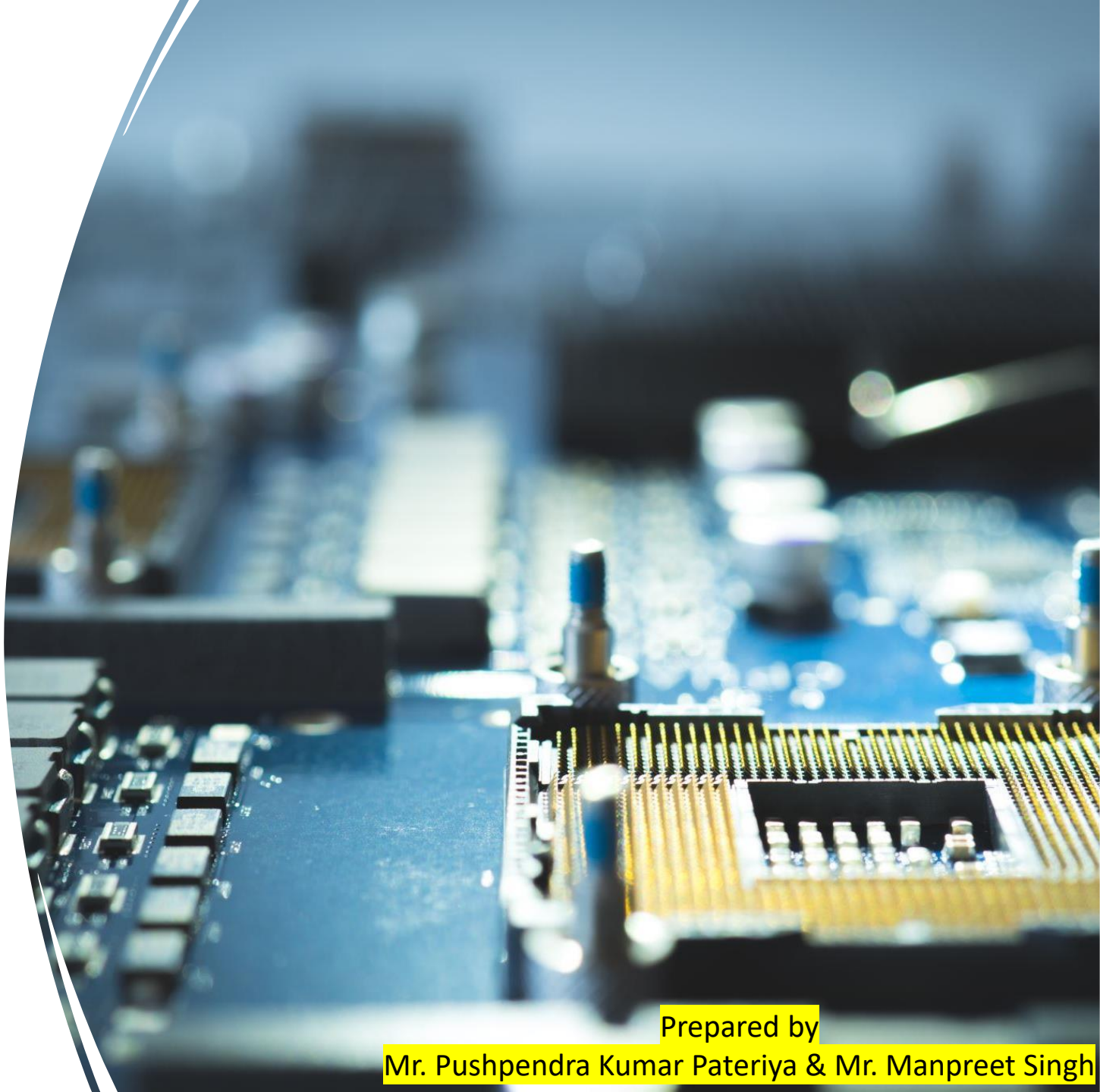
Many CLI programs use a feature called input/output redirection. How do they work?

- Input/output redirection is a powerful feature in command-line interfaces (CLI) that enables users to manipulate how input and output streams are handled for command execution. It allows the output of one command to become the input of another, and it can redirect output to files or from files to commands, among other functionalities.
- **Input Redirection:**
  - Standard Input (stdin) (<):
    - Redirects input from a file to a command.
    - Example: sort < input.txt sorts data from input.txt.
- **Output Redirection:**
  - Standard Output (stdout) (>):
    - Redirects the output of a command to a file or another command.
    - Example: ls > file.txt saves the output of ls into a file named file.txt.
- **Standard Error (stderr) (2> or &>):**
  - Redirects error messages to a file or another command.
  - Example: command_not_found 2> error.txt stores error messages in error.txt.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Can I take the output of one program and send it as the input of another one?

- Yes, you can use a feature called "piping" to take the output of one program and feed it as the input of another program in a command-line interface (CLI) environment.
- Piping is accomplished using the pipe symbol |.
- Here's an example: program1 | program2
- ls -l | grep '^d'

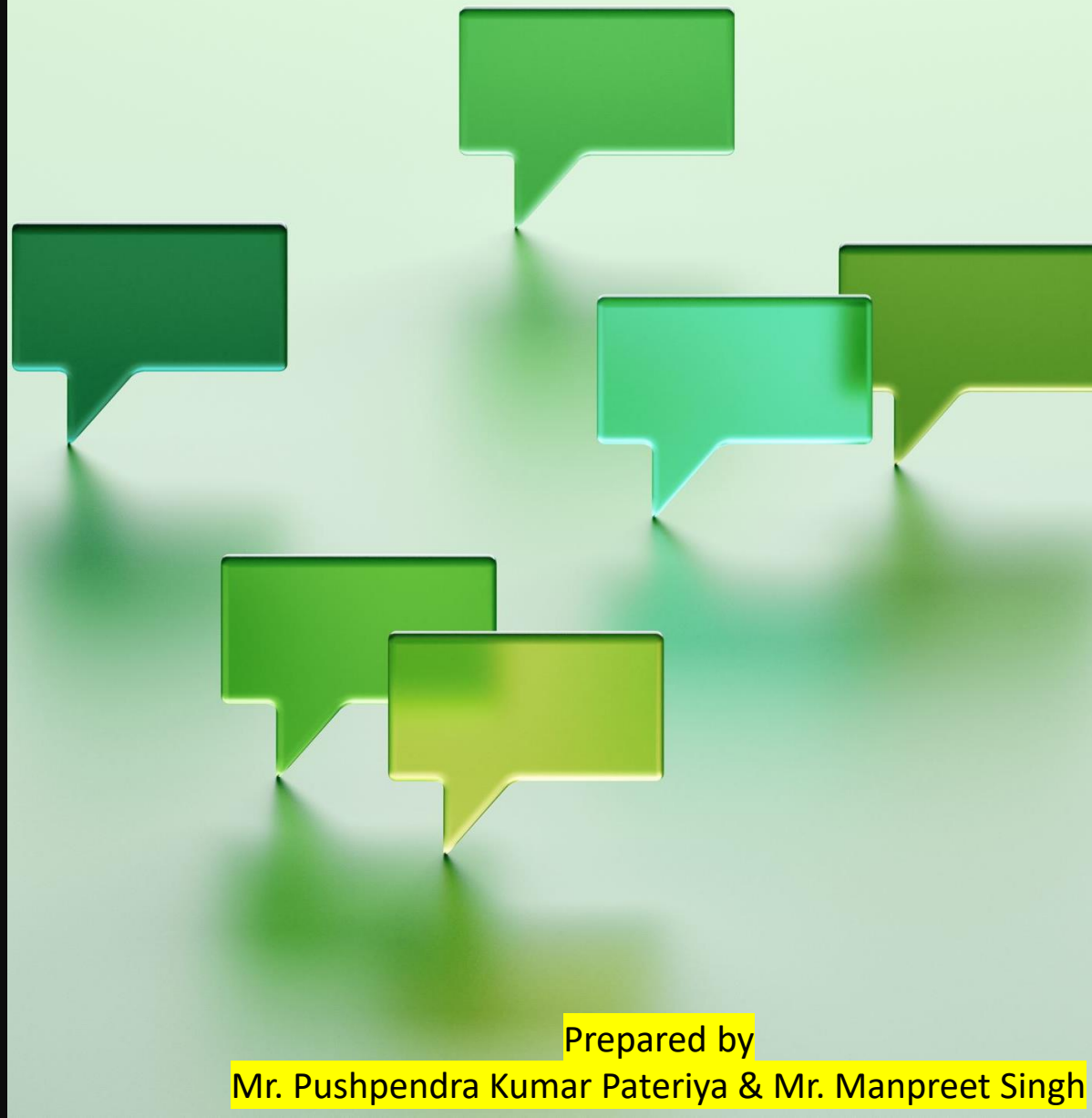


Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is a command substitution? How does it function?

- Command substitution is a feature in shell scripting and command-line interfaces that allows the output of a command or sequence of commands to replace the command itself. It enables capturing the output of a command and using that output as an argument or part of another command.
- In Unix-like systems, command substitution can be achieved using one of two syntaxes:

1. **Backticks (`...`):**
   - Older syntax for command substitution.
   - Example: result=`command`
2. **Dollar sign and parentheses ($(...)):**
   - More modern and recommended syntax for command substitution.
   - Example: result=$(command)



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Why should I opt to use vi editor?

Opting to use the vi editor (or its improved variant, vim) can be advantageous for several reasons:

- **Availability**: vi is available on most Unix-like systems by default, making it a ubiquitous text editor across various platforms.
- **Efficiency**: Once mastered, vi offers a highly efficient and powerful editing environment. Its modal interface allows for quick navigation, editing, and manipulation of text without needing to reach for the mouse.
- **Versatility**: vi is versatile and offers a wide array of commands for editing, searching, and replacing text, making it suitable for tasks ranging from quick edits to complex programming.
- **Customization**: vi is highly customizable. With vim (Vi IMproved), users can configure and extend its functionality using plugins, custom mappings, and scripting, tailoring the editor to their specific needs.
- **Lightweight**: vi is lightweight and doesn't require significant system resources, making it ideal for editing files even on resource-constrained systems or remote servers.
- **Stability**: vi has been around for decades and is known for its stability and reliability. Its consistent behavior across different systems ensures familiarity and ease of use.
- **Learning Curve**: Though vi might have a steeper learning curve initially, investing time in learning its commands and features can lead to increased productivity and efficiency in the long run, especially for users who frequently work in terminal environments.
- **Industry Standard**: In certain environments, especially in software development or system administration, familiarity with vi or vim is highly valued. Many professionals in these fields use vi as their primary text editor.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What are the basic vi commands?

Modes:

- **Normal Mode:** The default mode for navigation and issuing commands.
- **Insert Mode:** Allows text entry.
- **Command-Line Mode:** For entering commands like saving, quitting, searching, etc.

Editing:

- **x:** Delete the character under the cursor.
- **dd:** Delete the entire line.
- **u:** Undo the last change.
- **yy:** Copy (yank) the entire line.
- **p:** Paste the copied or deleted text after the cursor.
- **r:** Replace a single character under the cursor with a new one.
- **J:** Join lines, merging the current line with the next line.

Basic Navigation:

- **h, j, k, l:** Move the cursor left, down, up, and right respectively.
- **w, b:** Move the cursor forward or backward by a word.
- **0 (zero):** Move the cursor to the beginning of the current line.
- **$:** Move the cursor to the end of the current line.
- **G:** Move to the end of the file.
- **gg:** Move to the beginning of the file.

Saving and Quitting:

- **:w:** Save changes (write).
- **:q:** Quit if no changes are made.
- **:q!:** Quit without saving changes.
- **:wq** or **:x:** Save changes and quit.

Entering/Exiting Insert Mode:

- **i:** Enter Insert mode before the cursor.
- **a:** Enter Insert mode after the cursor.
- **o:** Open a new line below the current line and enter Insert mode.
- **O:** Open a new line above the current line and enter Insert mode.
- **Esc:** Exit Insert mode and return to Normal mode.

Searching:

- **/pattern:** Search forward for a specific pattern.
- **n:** Move to the next occurrence of the search pattern.
- **N:** Move to the previous occurrence of the search pattern.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

I want to find out the number of bytes and words in my file. How can I do this?

You can use various commands available in Unix-like systems (such as Linux) to find out the number of bytes and words in a file. Here are a few commands you can use:

- Using wc Command:
  - The wc (word count) command can provide the number of lines, words, and bytes in a file.
  - To count the number of bytes in a file: **wc -c filename**
- Using stat Command:
  - To get the size of the file in bytes: **stat -c %s filename**
- Using du Command:
  - To get the apparent size of the file in bytes: **du -b filename**

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

How can I translate characters from a standard input?

We can use the tr command in Unix-like systems to translate characters from standard input. The tr command is used to translate, squeeze, and delete characters.

- Basic Syntax: tr [OPTIONS] SET1 [SET2]
  - SET1: Defines the set of characters you want to translate from.
  - SET2: Defines the set of characters you want to translate to.

Examples:

- **Translate lowercase letters to uppercase:** echo "hello" | tr '[:lower:]' '[:upper:]'
- **Translate a specific character:** echo "abcde" | tr 'a' 'z'
- **Translate multiple characters:** echo "12345" | tr '123' 'abc'
- **Delete specific characters:** echo "Hello, World!" | tr -d '[:punct:]'
- **Squeeze repeated characters:** echo "hellooooooo" | tr -s 'o'

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is SED and how is it used?

SED stands for Stream Editor, and it is a powerful command-line tool used for text manipulation in Unix-like operating systems. It is designed for processing text files, applying transformations or edits to text content based on specified commands or patterns.

Features and Usage of SED:

- Pattern Matching and Substitution: SED allows finding patterns in text and performing substitutions.
  - Example: sed 's/old_text/new_text/g' filename replaces all occurrences of old_text with new_text in the file filename.
- Text Filtering and Selective Printing: SED can filter lines and selectively print specific lines or ranges of lines.
  - Example: sed -n '10,20p' filename prints lines 10 to 20 from filename.
- Inserting and Appending Text: SED can insert or append text at specific locations or lines in a file.
  - Example: sed '3i\Insert this line' filename inserts Insert this line before line 3 in filename.
- Deleting Lines: SED can delete lines based on patterns or line numbers.
  - Example: sed '/pattern/d' filename deletes lines containing pattern in filename.
- Regular Expressions: SED supports powerful regular expressions for pattern matching and text manipulation.
  - Example: sed 's/[0-9]*/& &/' filename duplicates numbers in filename.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

I want to carry out a lengthy task in the background while using the terminal for another purpose. Is it possible to selectively stop or suspend the execution of some processes?

Yes, you can selectively stop or suspend the execution of processes in a Unix-like operating system using the Ctrl + Z keyboard shortcut to suspend a running process and the bg, fg, or kill commands to manage processes in the background or foreground.

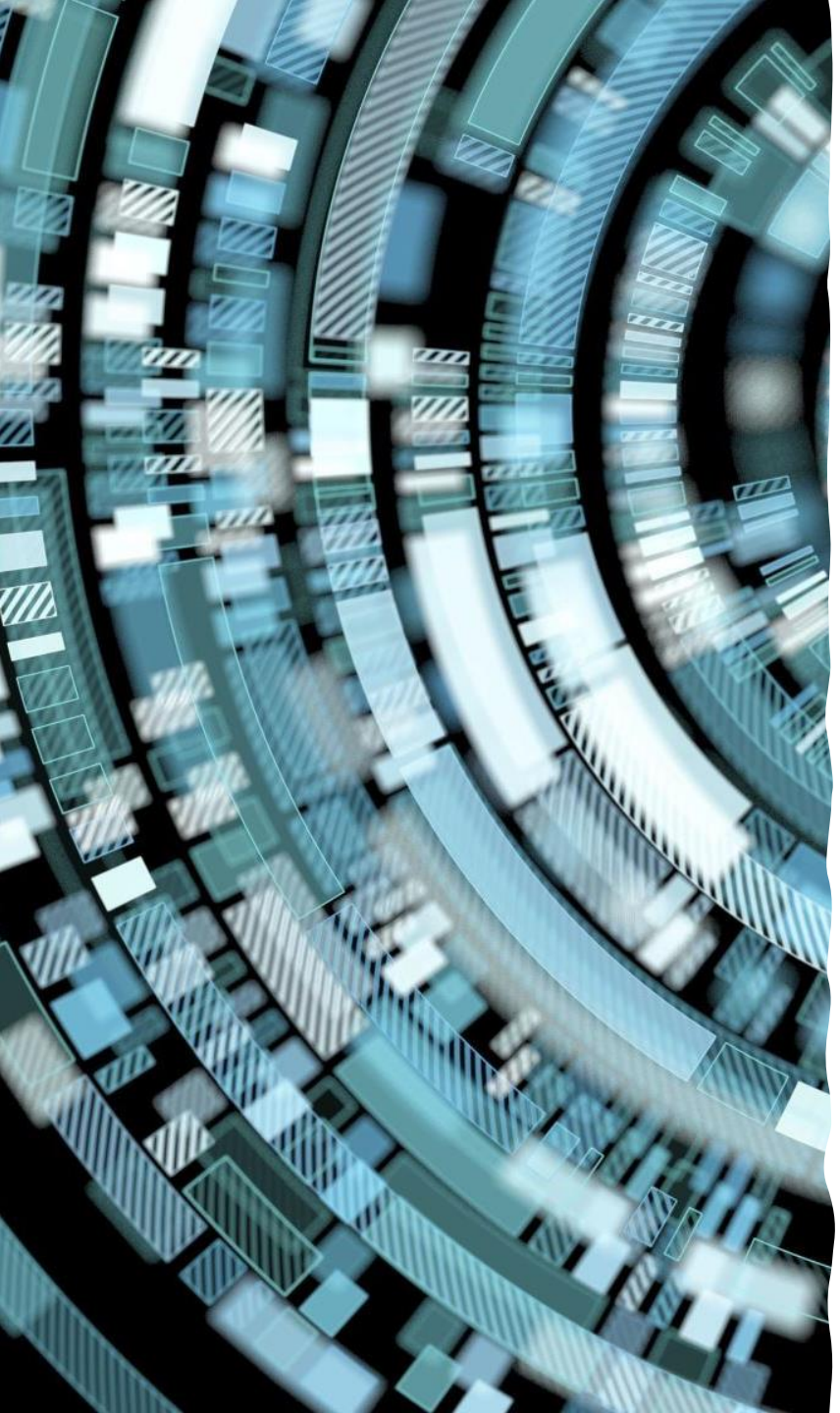
Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

How should I view and search in the man pages?

Searching within Man Pages:

- Search Forward:
  - While viewing a man page, press / followed by the search term, then press Enter.
  - Example: /search_term and press Enter.
- Search Backward:
  - Similar to searching forward, but use ? instead of /.
  - Example: ?search_term and press Enter.
- Navigating through Search Results:
  - After searching, use n to move to the next occurrence of the search term or N to move to the previous occurrence.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh



What exactly is /proc file system in LINUX? How does it work?

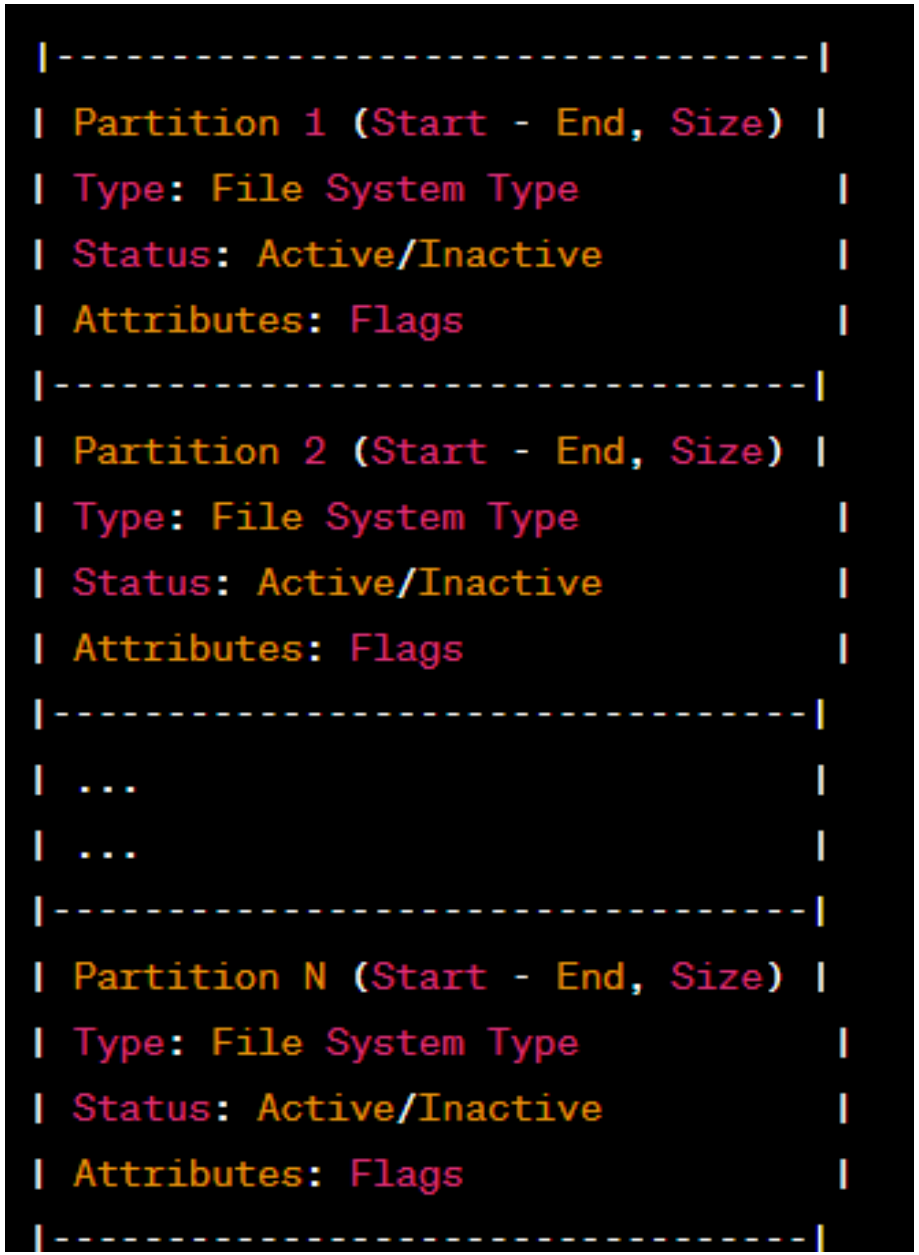
- The /proc filesystem serves as a dynamic interface between the kernel and user space, offering a convenient way to access a wide range of system-related information and kernel data structures, aiding in system monitoring, troubleshooting, and system configuration.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What does a partition table look like?

A partition table is a data structure on a storage device (like a hard drive or SSD) that defines how the storage space is divided into separate sections known as partitions. It contains information about the layout and characteristics of each partition on the disk.

- Structure of a Partition Table:
  - A partition table typically includes the following information for each partition entry:
  - Partition Number (ID): An identifier for each partition.
  - Starting LBA: The starting address of the partition in LBA (Logical Block Addressing).
  - Ending LBA: The ending address of the partition.
  - Size: The size of the partition in sectors or blocks.
  - Partition Type: An identifier specifying the type of data contained in the partition (e.g., NTFS for Windows, ext4 for Linux).
  - Partition Status: Information about whether the partition is active or bootable.
  - Attributes: Additional attributes or flags associated with the partition, such as the file system type, bootable flag, etc.
- Visual Representation:



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is the advantage of partitioning my disk?

Partitioning a disk offers several advantages:

- Organization and Management: Partitioning allows you to organize and manage your data more efficiently by creating separate sections on the disk for different purposes. For example, you can have separate partitions for the operating system, applications, user data, and backups.
- Isolation and Protection: By segregating data into different partitions, you can isolate critical system files from user data. This isolation helps prevent accidental deletion or corruption of essential system files when working with user data.
- Multi-boot Configurations: Partitioning enables the creation of multiple partitions, allowing you to set up different operating systems on the same disk. Each OS resides in its partition, enabling a multi-boot setup without interference between the operating systems.
- Performance Optimization: Partitioning can aid in performance optimization. For instance, placing frequently accessed files or swap partitions on a separate partition or on a specific part of the disk can enhance performance.
- Backup and Recovery: When backing up data, having different partitions makes it easier to back up specific sections selectively. It simplifies backup procedures, reduces backup time, and enables quick restoration of specific data in case of issues.
- Security and Encryption: Partitioning can be useful for implementing security measures. For instance, you can encrypt specific partitions containing sensitive data, adding an extra layer of security to protect your information.
- Disk Maintenance and Troubleshooting: Partitioning aids in disk maintenance and troubleshooting. It allows you to perform specific operations (such as formatting, resizing, or checking for errors) on individual partitions without affecting other partitions or the entire disk.
- File System Flexibility: Partitioning enables the use of different file systems on separate partitions. For instance, you can have one partition formatted with NTFS for Windows compatibility and another with ext4 for Linux.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is MBR?

MBR stands for Master Boot Record. It's a crucial data structure located at the very beginning (sector 0) of a storage device such as a hard drive, solid-state drive (SSD), or other similar storage media.

- Components of the Master Boot Record (MBR):
  - Bootloader: The MBR contains a small piece of executable code known as the bootloader. This code is responsible for initiating the boot process of the operating system.
  - Partition Table: The MBR also contains a partition table, which is a table that stores information about the disk's partitions, such as their sizes, starting and ending points, and partition types. In the traditional MBR layout, the partition table can hold information for up to four primary partitions or three primary partitions and one extended partition with multiple logical partitions within it.
- Functions of the Master Boot Record (MBR):
  - Bootstrapping: When a computer boots up, the BIOS (or UEFI firmware) loads the MBR into memory. The bootloader code within the MBR then gets executed, initiating the process of loading the operating system.
  - Partition Information: The MBR's partition table holds crucial information about how the disk is partitioned. It contains the necessary details for the system to locate and access the partitions where operating systems and data are stored.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is hard drive partitioning?

Hard drive partitioning is the process of dividing a single physical hard drive into separate sections or partitions, each of which functions as if it were an independent drive. Each partition appears as a distinct volume or drive letter in the operating system, allowing users to organize and manage their data more efficiently.

Purpose of Hard Drive Partitioning:

- Organizing Data: Partitioning allows users to organize data by separating different types of files, operating systems, or user data into distinct sections, making it easier to manage and locate specific information.
- Multi-boot Configurations: Partitioning enables the installation of multiple operating systems on the same hard drive, allowing users to have a dual-boot or multi-boot setup with different operating systems residing in their respective partitions.
- Security and Isolation: By segregating user data from system files or separating critical data from non-critical data, partitioning can enhance security and protect sensitive information from accidental deletion or corruption.
- Backup and Recovery: Having separate partitions can simplify backup procedures. Users can back up specific partitions or volumes selectively, making backup and recovery processes more efficient.
- Performance Optimization: Partitioning can aid in performance optimization by placing frequently accessed files, swap partitions, or specific data on dedicated partitions, potentially improving system performance.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What does a bootloader do?

A bootloader is a small program that resides in a specific region of a storage device (like a hard drive, SSD, or a similar media) and is responsible for initiating the operating system's startup process. Its primary function is to load and execute the operating system kernel to begin the system's boot sequence.

Functions of a Bootloader:

- Initialization and System Startup: When a computer is powered on or restarted, the BIOS (or UEFI firmware) initiates the boot process by loading the bootloader from the designated boot device (typically the hard drive or a specific partition).
- Loading the Operating System Kernel: The bootloader's main task is to locate and load the operating system kernel into memory. It reads the necessary files or data required to start the operating system.
- Configuration and Environment Setup: Bootloaders often provide options to select and boot into different operating systems (in the case of multi-boot configurations) or booting into recovery or diagnostic modes. Bootloaders may load configuration files or parameters necessary for the kernel to start properly.
- Error Handling and Recovery: Bootloaders handle error conditions during the boot process, such as system crashes, corrupted boot files, or missing operating system files. They may provide options for recovery or repair.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

How do I determine the type of an unfamiliar file on the Linux system?

- We can determine the file type using file command.
- File Command: The file command is used to determine the type of a file by examining its contents or metadata.

  Syntax:

  file <file_name>

  file myfile.txt

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Why To Learn Linux?

Linux is a UNIX based operating system.

Linus Torvalds first introduced it.

It is an open source operating system

Open Source *means you can download the file and change the code as you like.*

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What Makes Linux Great?

- Free and Open Source
- More Secure
- Multiprogramming System
- Provides both CLI and GUI
- Multi-user Access

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What are basic elements or components of Linux?

Application Programs: A set of functions designed to perform a set of tasks.

Shell: It acts as an interface between the user and the Kernel.

Kernel: It is the core component of Linux, it acts as an interface between software and hardware.

Hardware layer – Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Unix vs Linux

UNIX

- Default shell is Bourne Shell(sh).
- Linux provides two GUIs, KDE and Gnome.
- Linux is free.

LINUX

- Default shell is Bourne Again Shell (BASH).
- Initially, Unix was a command-based OS, however later a GUI was created called Common Desktop Environment.
- Unix is not totally free

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Describe the root account

The root account is like a systems administrator account and allows you full control of the system.

It is represented with #

```
[root@host ~]#
```

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

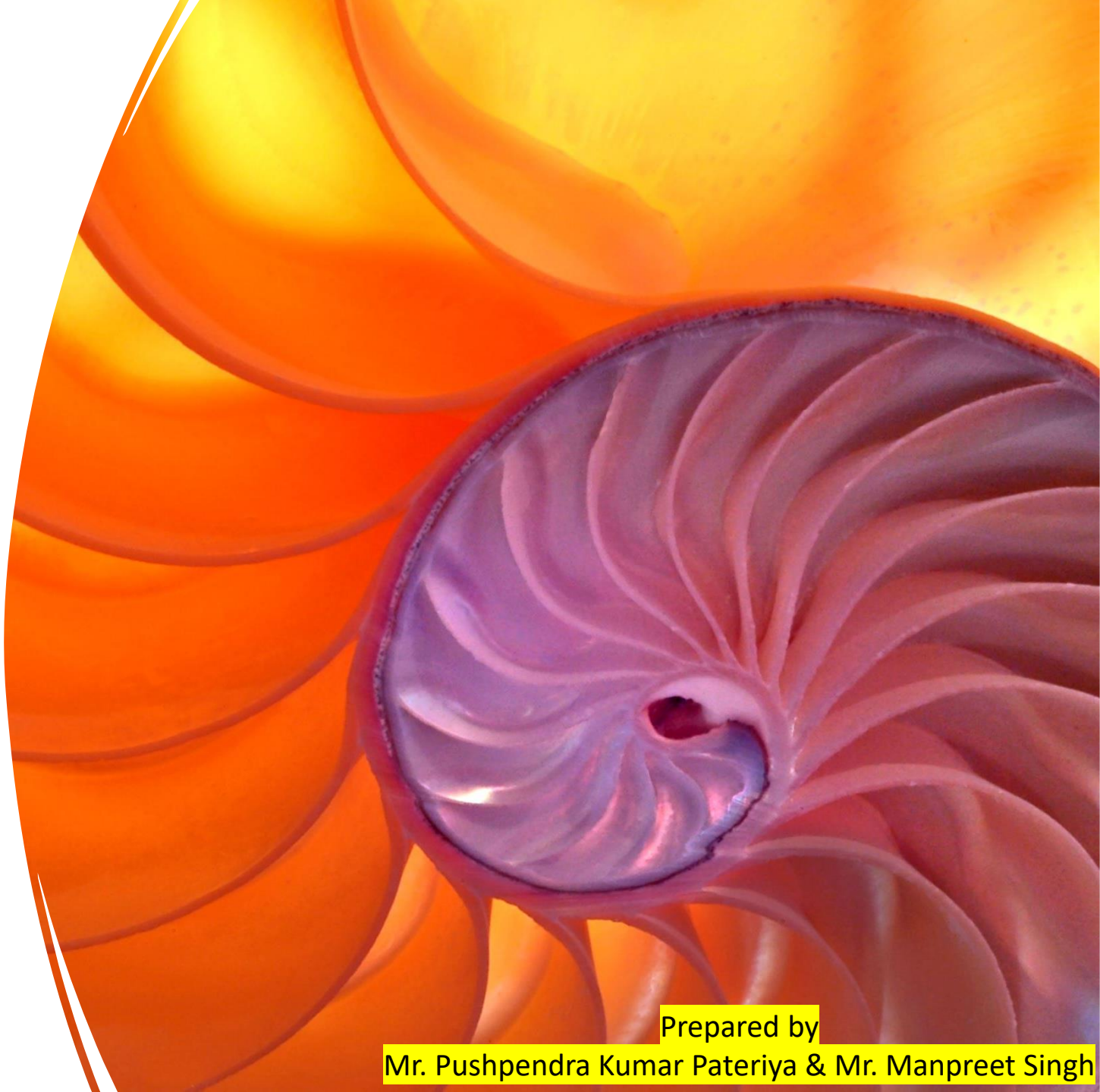
Describe normal account

- This is the regular user account.
- After the installation, we can create as many regular user accounts as we need.
- This account has moderate privilege.
- As per requirement, it can be disabled or deleted.
- By default represented with ‘$’.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is Linux Shell?

- Linux Shell is called the user interface available between the kernel and the user.
- Linux shell takes human-readable commands as input and transforms them into a language that is kernel understandable.
- Types
  i. Bourns Shell
  ii. ZSH
  iii. TCSH
  iv. Bourns Again Shell or BASH
  v. Korn Shell or KSH
  vi. C Shell or CSH



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is interactive and non-interactive shell?

Interactive Shell

- /bin/bash and /bin/sh

Non-interactive shell

- /sbin/nologin

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Name some Linux Distributions

Ubuntu

Debian

CentOS

Fedora

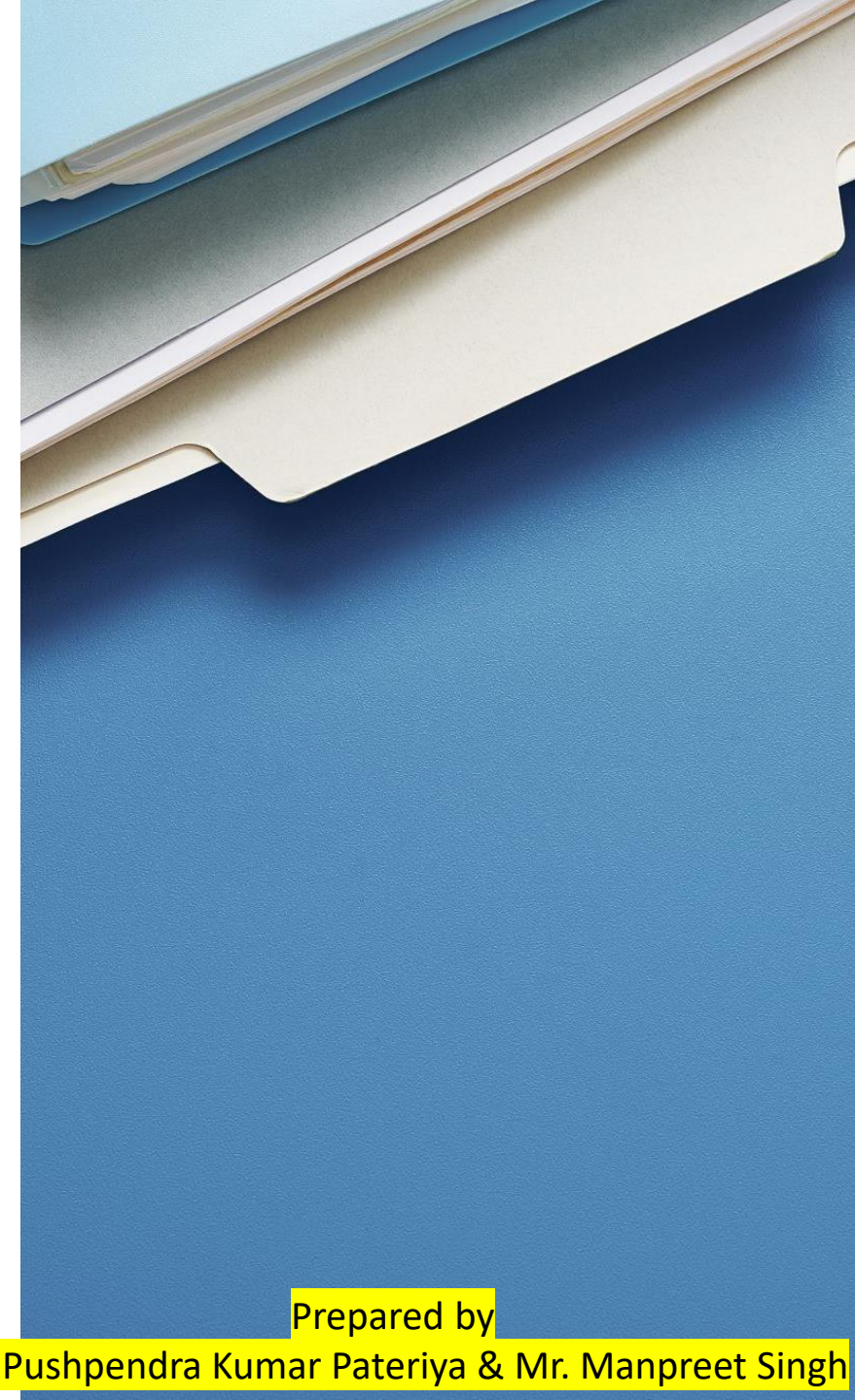
RedHat

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Explain File Permissions types in Linux?

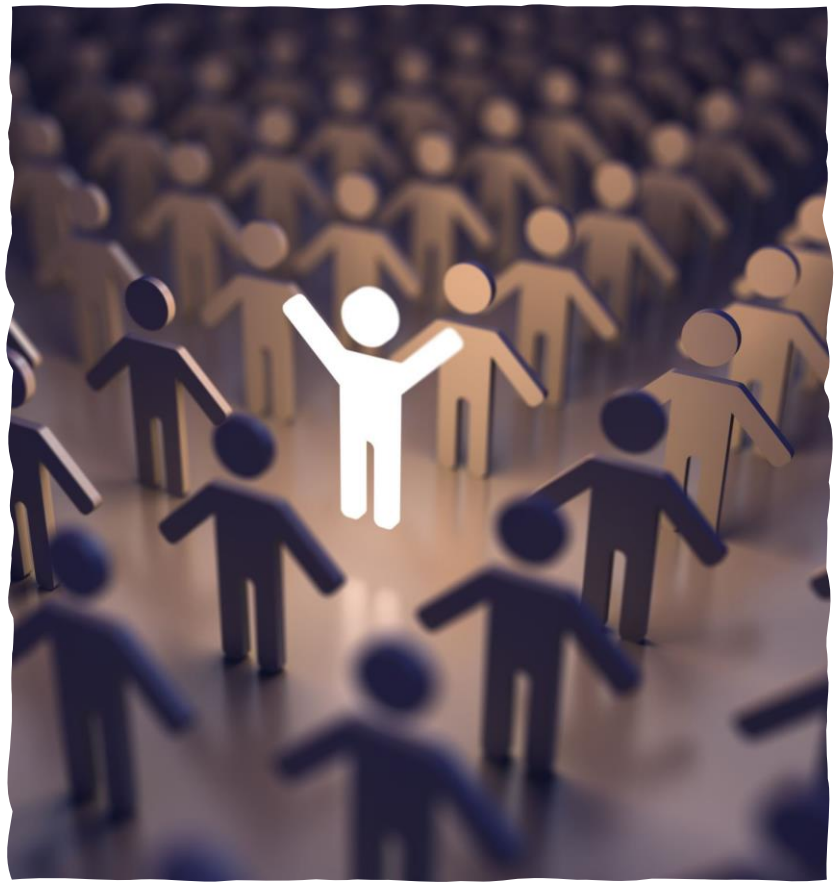
Each file or directory has 3 permissions:

- **Read:** It refers to only they can read the file.
- **Write:** It refers that they can write the file or modify the file of a directory.
- **Execute:** It affects the user's capability to execute the file or to view the file of a directory.



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Explain File Permission groups in Linux?



There are three user-based permission groups for each file and directory:

- **Owner:** Owners only will have to access the file or directory, they will not impact the actions of other users.
- **Group:** These permissions apply only to the group, that has been assigned to the file or directory. They will not impact the actions of other users.
- **All Users:** These permissions are applied to all users on the system.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

If you have saved a file in Linux. Later you wish to rename that file, what command is designed for it?

mv

Example

mv apple.txt grapes.txt

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is the absolute and relative path?

- **Absolute path** is the full path of the directory. It always starts with “/” .
- Example:
  - cd /var/tmp/abrt/
- **Relative path** is necessary from current location to reach particular directory doesn’t start with “/”.
- Example:
  - cd .. , cd –

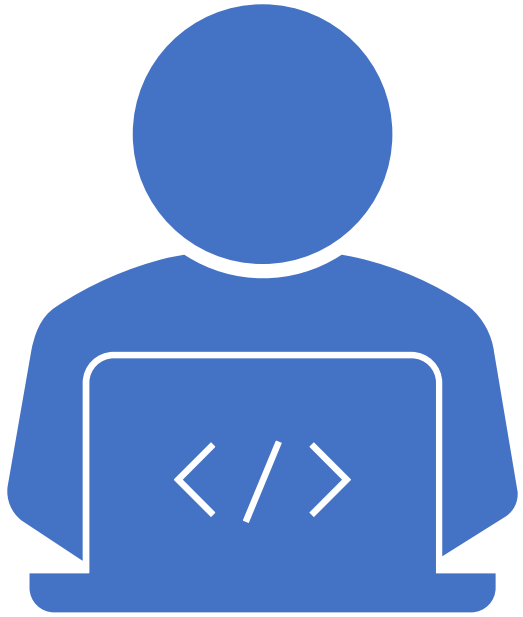
Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What are link and it's types?

- Links act similarly to shortcuts in Windows.

- Types:
  - Hard Link (ln)
    - $ ln [original filename] [link name]
  - Soft/Symbolic Link (ln -s)
    - $ ln -s [original filename] [link name]
- Difference:
- Deleting the original file does not affect the hard link but Deleting the original file makes the soft link inactive.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh



What are the basic commands for user management?

- Useradd-> create a new user account
- Example: useradd Nikhil
- Usermod->modify an existing user account
- Example: usermod –u Nikhil 1111
- Userdel-> delete an existing user account
- Example: userdel Nikhil

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Which file is used to store local users' information?

/etc/passwd	❶user01:❷x:❸1000:❹1000:❺User One:❻/home/user01:❼/bin/bash
1. username	
2. password	
3. uid	
4. gid	
5.comment field	
6. home directory	
7. default shell	

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

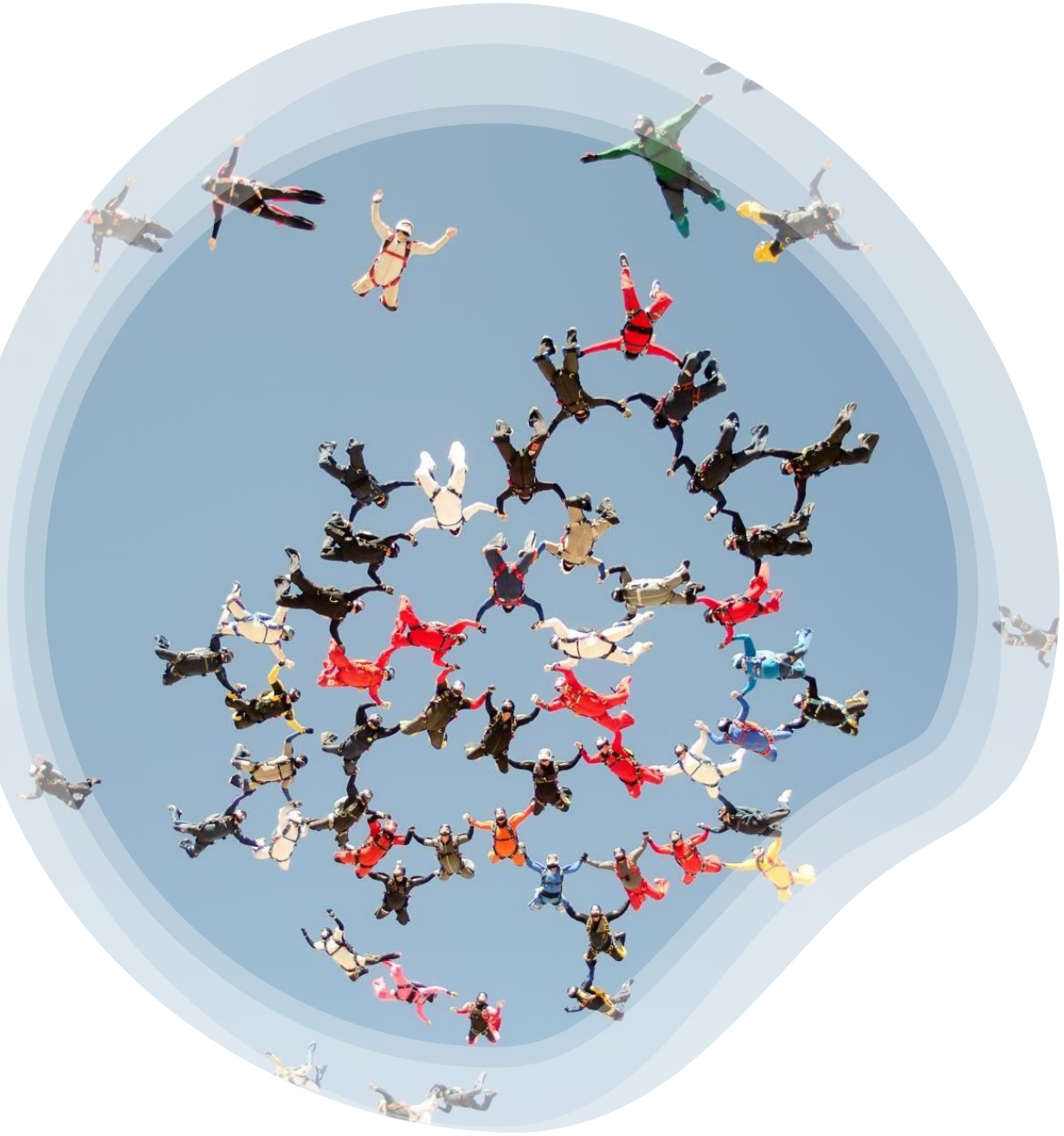
What are the basic commands for group management?

- groupadd->create a new group
- Example: groupadd LPU

- groupmod->modify an exiting group
- Example: groupmod –g 3333 LPU

- groupdel->delete a group
- groupdel LPU

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh



Which file is used to store local groups' information?

/etc/group

```
1group01:2x:310000:4user01,user02,user03
```

1. groupname
2. password field
3. gid
4. member list

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is the significance of ‘passwd’ command?

- passwd command in linux is used to change the user password or login password.
- **Syntax**

  passwd [OPTIONS] [LOGIN]

Example

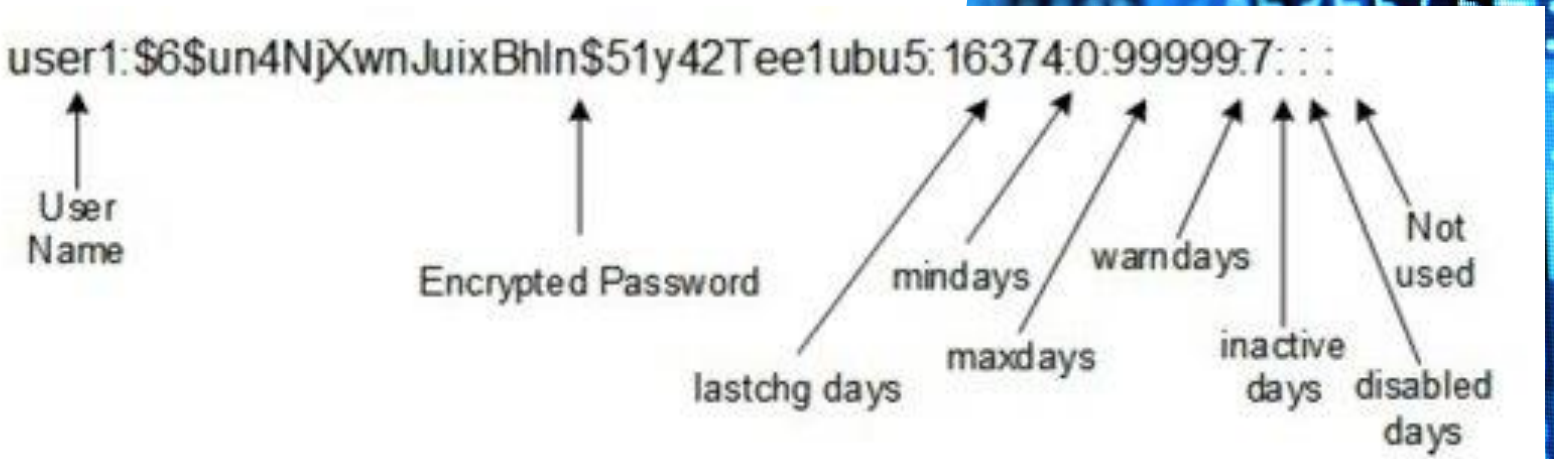
  passwd –d anil

-d -> to deletes password for user anil

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Which file stores users' password and password related information.

/etc/shadow



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Explain about chmod command?

- This command is used to change the permission of files and directories.

- Syntax: $ chmod permissions <file name>
- Example: $ chmod 756 fruits

u -> having rwx permissions

g -> having rx and no w permission

o -> having rw and no x permission

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

How to set default file permissions for files and directories?

- umask

- Example for file
- umask 025

-rw-r---w-

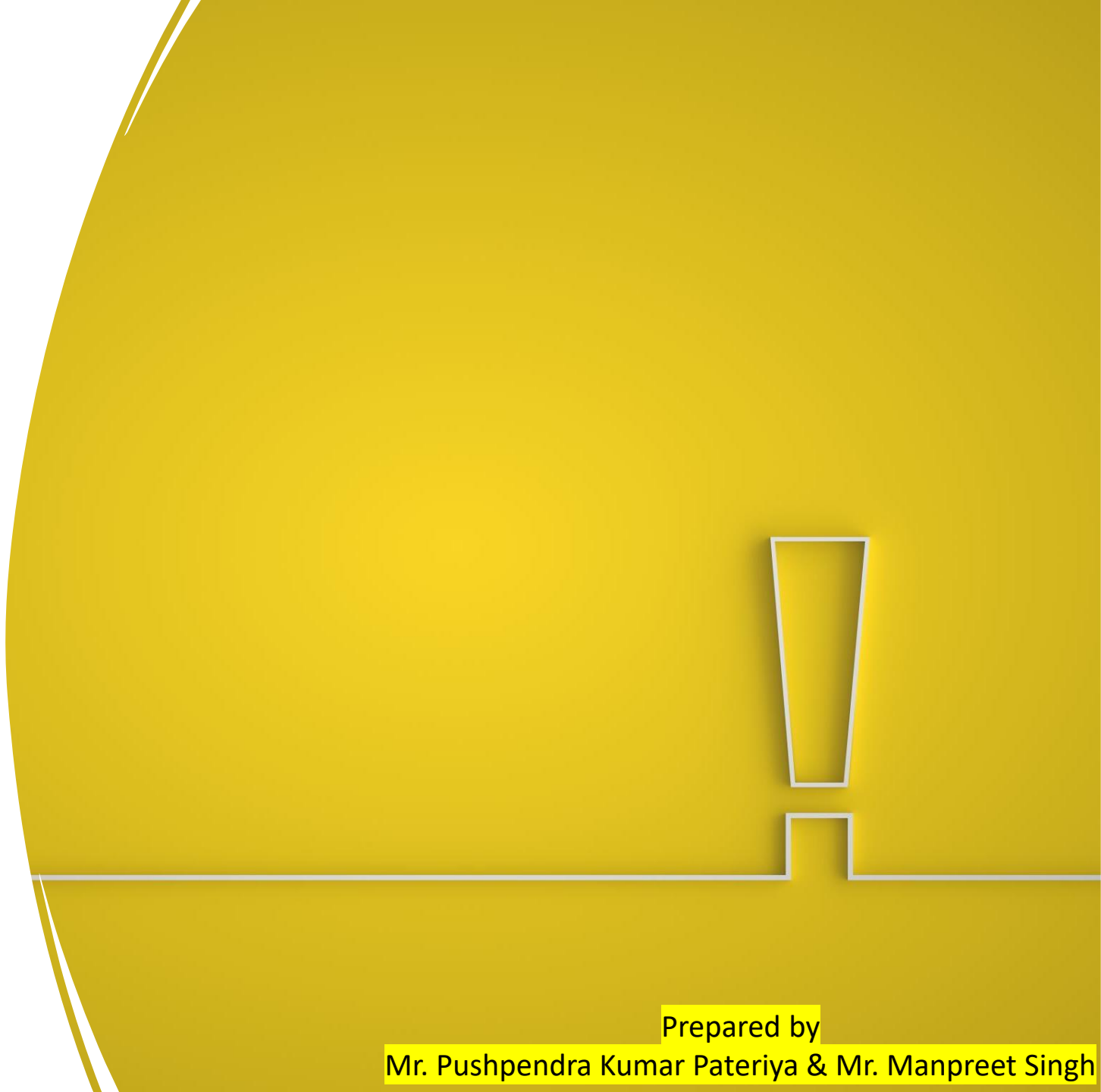
Example for directory

drwxr-x-w-

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Explain “s” permission bit in a file?

- “s” bit also called “set user id”(SUID) bit. “s” on file causes the process to have the privileges of the owner of the file during the instance of the program.



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is grep command in Linux?

grep command is a filter that is used to the global search for regular expressions.

Syntax: $ grep [options] pattern [files]

Examnple: $ grep –c ‘apple’ fruits.txt

-c → displays the count of the matching patterns.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

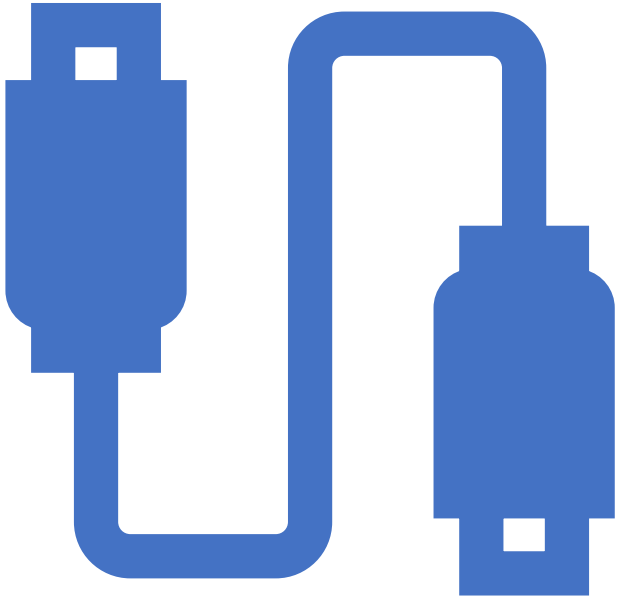
What are the different modes when using the vi editor?

- There are three kinds of modes in vi editors. They are :
  - Command Mode/ Regular Mode
  - Insertion Mode/Edit Mode.
  - Ex Mode/ Save Mode.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What do you understand about the standard streams?

- Stdin (standard input)->to receive an input.(keyboard)
- stdout(standard output)-> to send an output.(Terminal)
- stderr (standard error)-> to send an error. (Terminal)

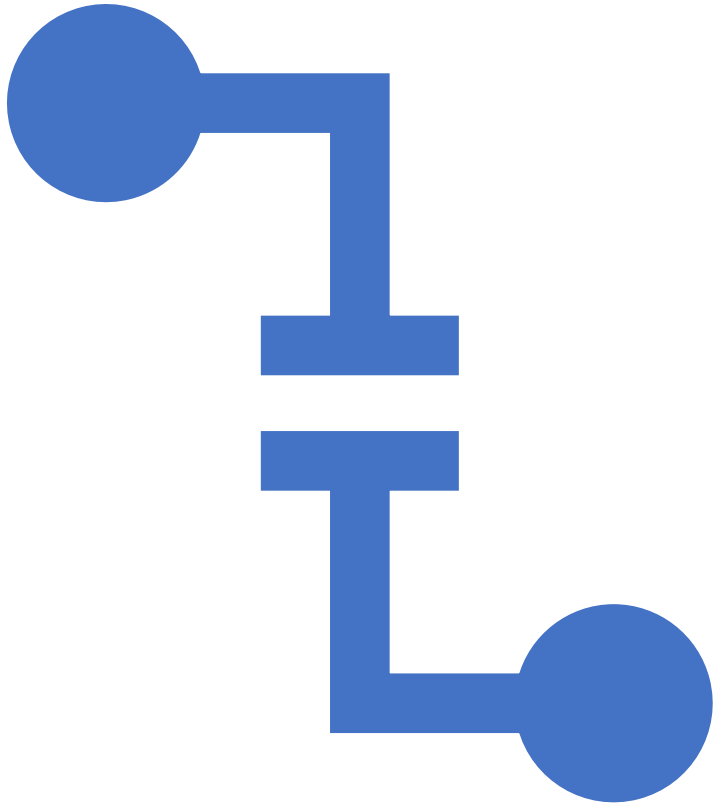


Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is redirection in Linux?

Sending output of process to a file

1. > -
2. >>
3. 2>



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What are inode and process id?

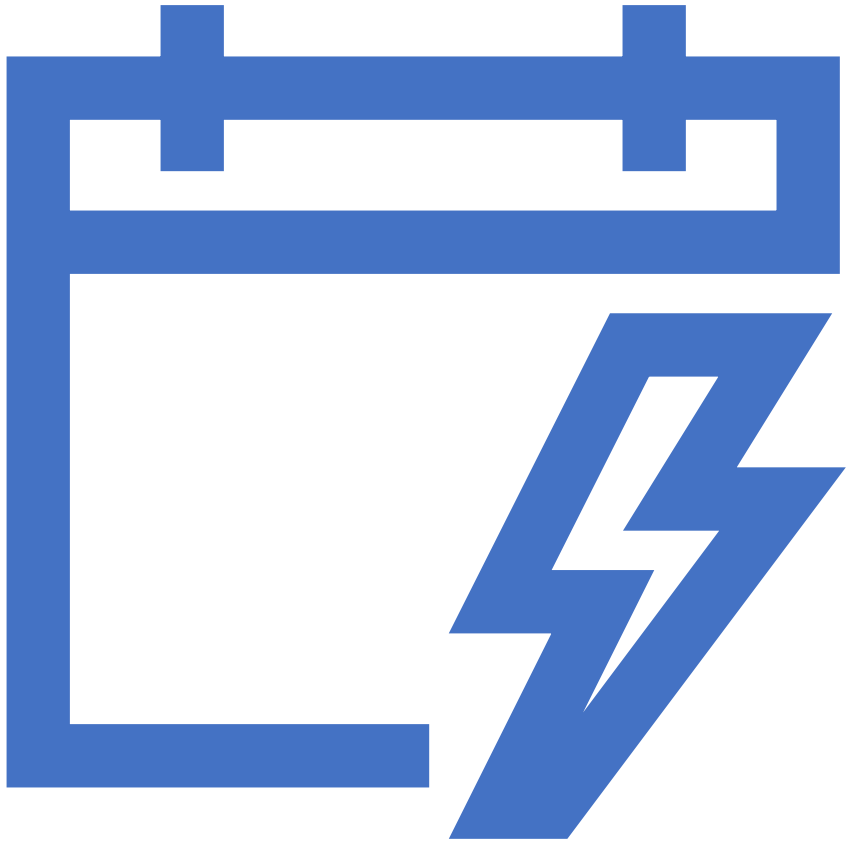
- The inode is a unique name given to each file.
- Command used: ls – i <filename>

- The process id is a unique name given to each process.
- Command used: top , ps

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

How to schedule recurring tasks?

- crontab
  - Syntax:
- crontab –e (to edit)
- crontab –l (to list scheduled task)
- crontab –r (to remove)
- Example:
- 8 20 24 1 * date>date.txt
- The above job means : The system date will saved in the date.txt file on 24th Jan, at 8:08PM every year



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is the job format for crontab

- Every job consists of six fields. The fields are:
  - Minutes (0-59)
  - Hours (0-23)
  - Day of Month (1-31)
  - Month (1-12)
  - Day of week (0-7 : 0 and 7 both mean Sunday)
  - Command

  When all the first five fields match the current data and time, the command in the sixth field is executed.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

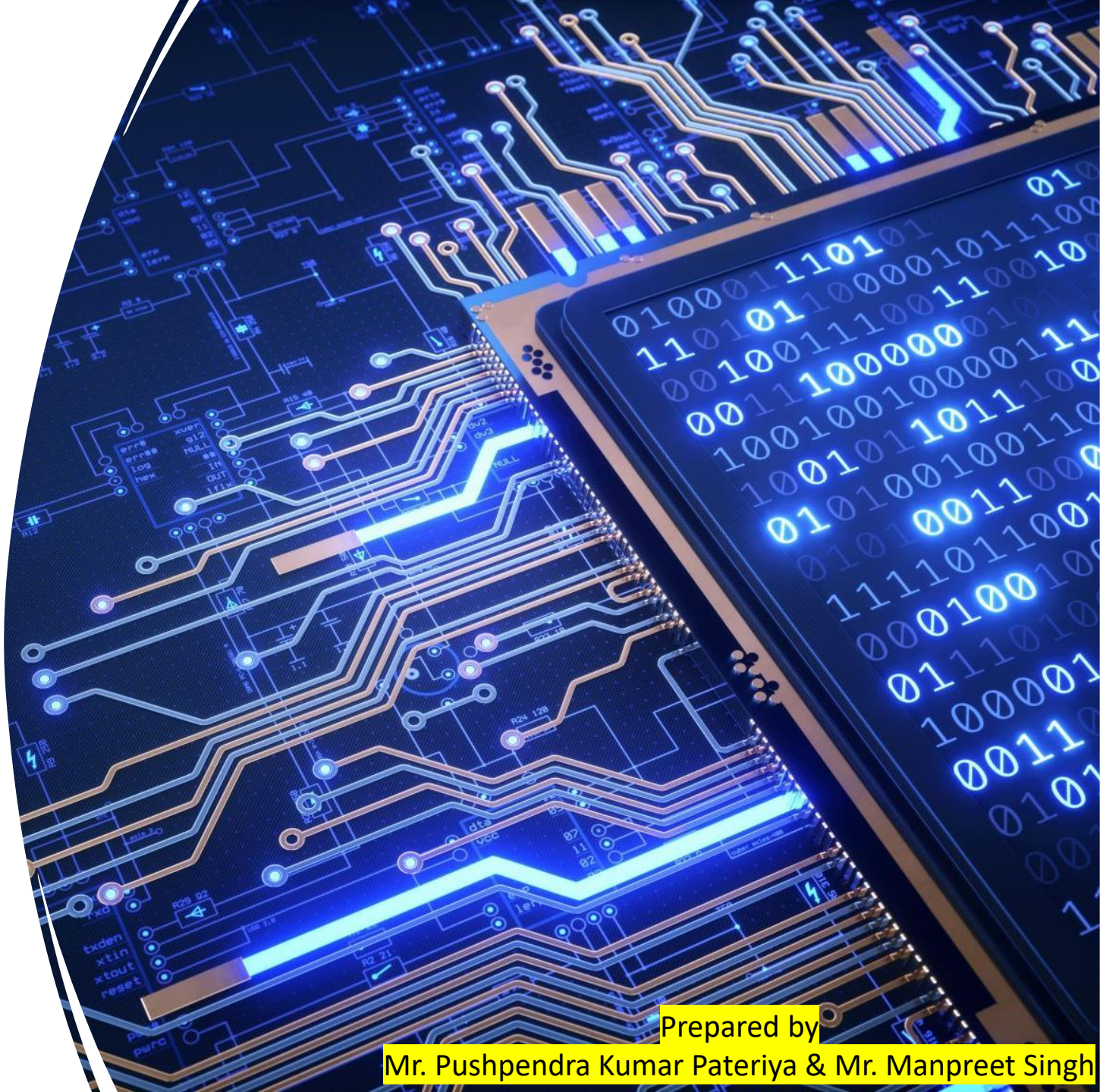
What is a File system in Linux?

- Linux file system stores and handles the data. Without a file system, it cannot know where the file starts from and where the file ends.
- In Linux, there are many file systems:
- Ext, Ext2, Ext3, Ext4, XFS, swap

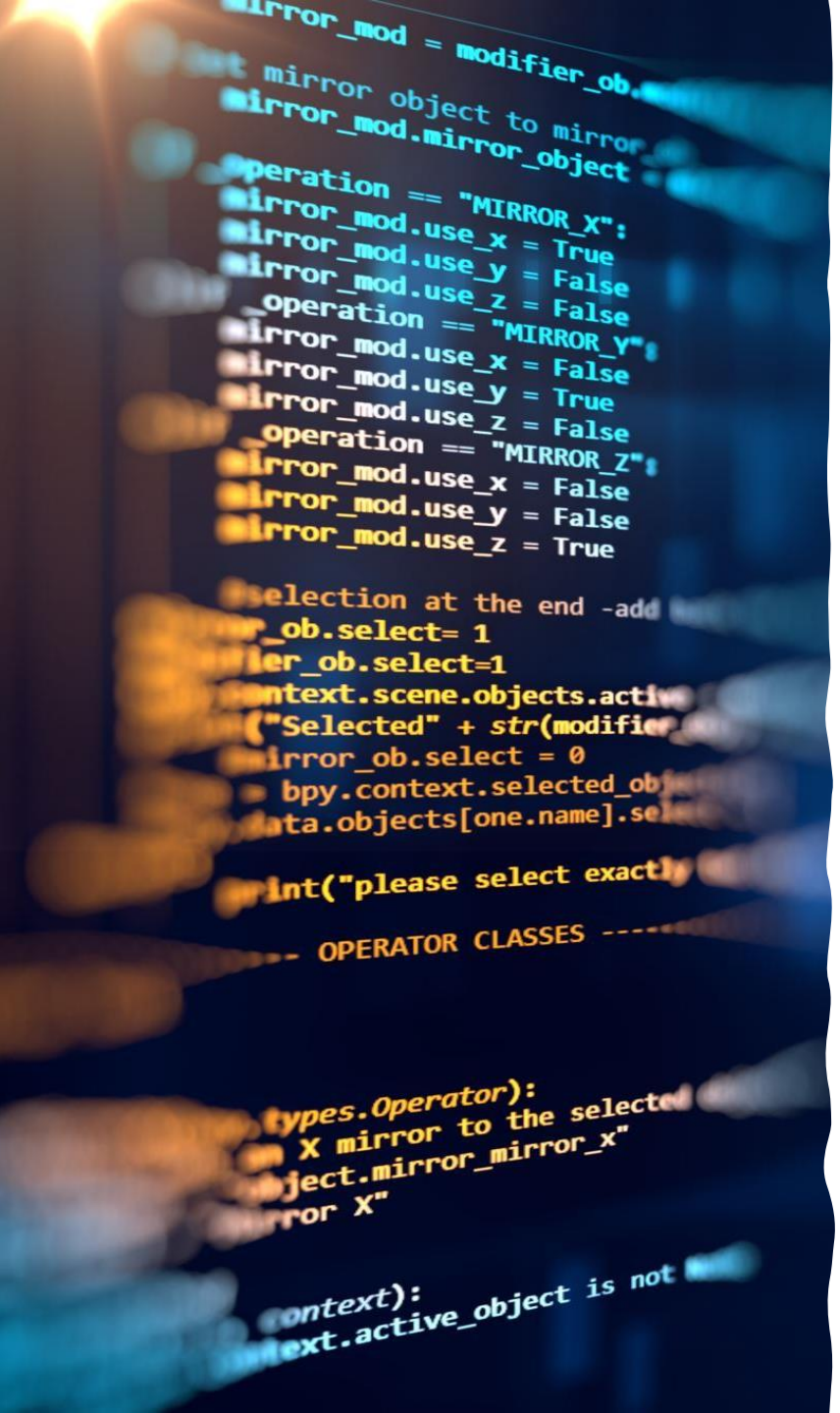
Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is a swap space?

- ***Swap space*** is a space on a hard disk that is a substitute for physical memory.
- It is used as virtual memory, which contains process memory images.
- Whenever our computer runs short of physical memory, it uses its virtual memory and stores information in memory on a disk.



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh



Why Shell scripting?

- *Shell script is a series of command(s) stored in a plain text file.*

- A script helps to automate tasks.

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

How to create a script?

- Three steps:

i. Create a script

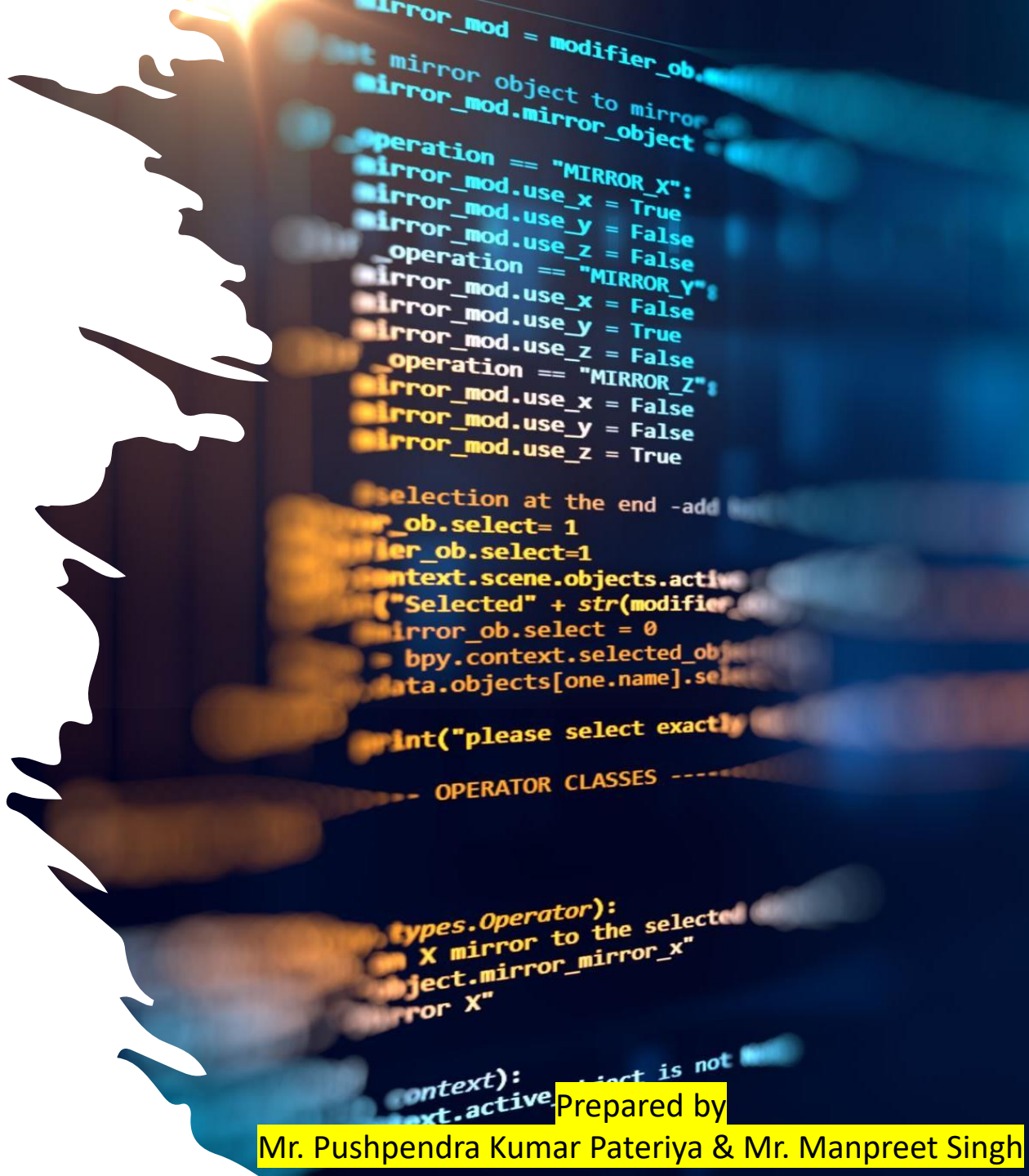
vi script-name

ii. Setup executable permission

chmod +x script-name

iii. *Run a script*

./script-name

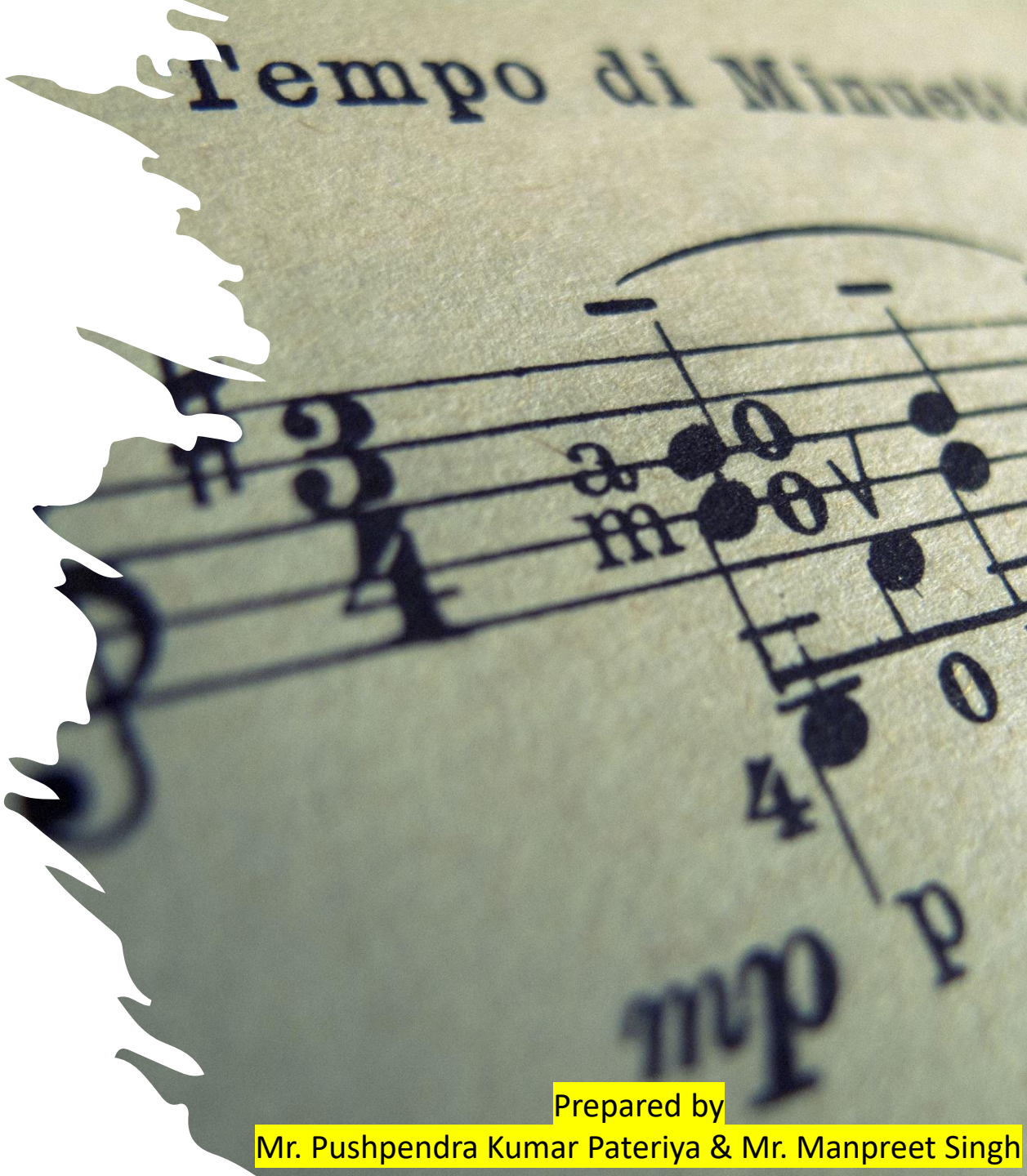


Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is the purpose of the shebang line?

- The **Sha-bang** or **She-bang** or **hash-bang** or **pound-bang** is the first line of a script which tells the system which program to use to interpret the commands in the script.

- For shell script it is not mandatory.
- Example: #!/bin/bash



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What are the types of variables used in shell scripting?

- **System variables/ Environmental Variables**
- Created and maintained by Linux itself. This type of variable defined in CAPITAL LETTERS.
- Example: echo $SHELL, echo $HOME,echo $USER etc.
- **User defined variables (UDV)** – Created and maintained by user. This type of variable defined in lower letters.
- Example: *x=10*

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What are positional parameters?

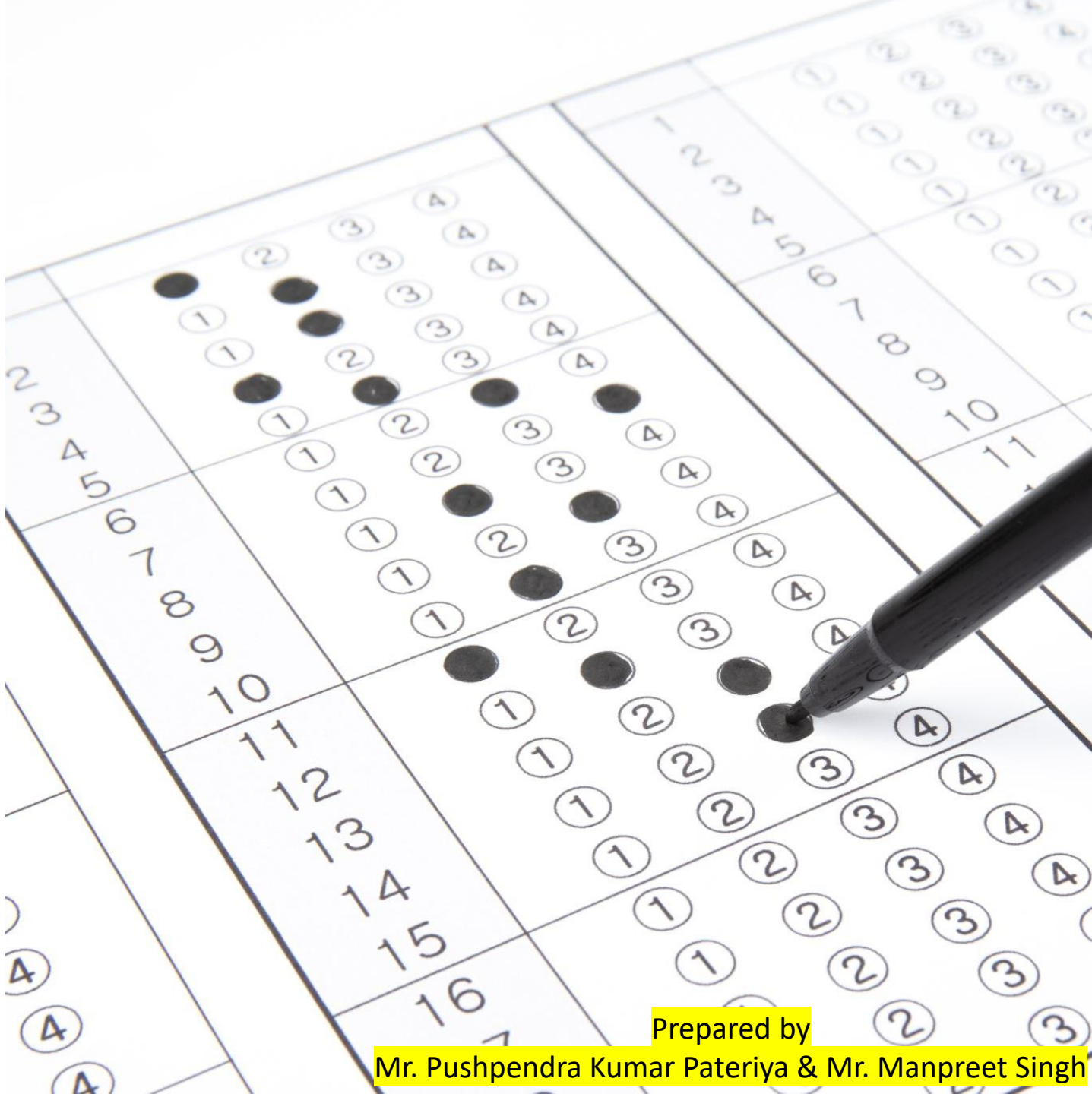
- Arguments passed through the command line.
- The value of these command line arguments is stored as below:
  - $0 – stores the name of the script
  - $1 – stores the first argument
  - $2 – stores the second argument
  - $3 – stores the third argument
  - $# – stores the total number of argument
  - $* – stores the value of all argument

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

How to get input from user?

- read statement

- Syntax:
  - read variable-name
  - Example
  - read fname



Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Which are relational operators in shell scripting?

-eq (equals to)

-lt (less than)

-le (less than equals to)

-gt (greater than)

-ge (greater than equals to)

-ne (not equals to)

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Explain different file test operators?

- -d ->Checks if file is a directory
- -f ->Checks if file is an ordinary file
- -r ->Checks if file is readable
- -w ->Checks if file is writable
- -x ->Checks if file is executable
- -s ->Checks if file has size greater than 0
- -e ->Checks if file exists

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

What is the purpose of ‘test’ command?

- *test command or [expr]* is used to see if an expression is true, and if it is true it return zero(0), otherwise returns nonzero(>0) for false.
- *Syntax: test expression OR [expression]*

Prepared by
Mr. Pushpendra Kumar Pateriya & Mr. Manpreet Singh

Thank you
Good Luck

https://www.linkedin.com/in/pushpendra-pateriya-344b5727/