**Practical 2 - Bash Commands Explanation**

**Commands and Explanations**

1. **$ date**
   * Displays the current date and time in the default format.
2. **$ date "+%y"**
   * Shows the last two digits of the current year (e.g., 24 for 2024).
3. **$ date "+%b"**
   * Displays the abbreviated name of the current month (e.g., Nov).
4. **$ date "+%m"**
   * Outputs the current month as a number (e.g., 11 for November).
5. **$ date "+%d"**
   * Shows the current day of the month (e.g., 20).
6. **$ date "+%j"**
   * Displays the day of the year (e.g., 324 for November 20).
7. **$ date "+%d-%b-%y"**
   * Outputs the date in the format: DD-MMM-YY (e.g., 20-Nov-24).
8. **$ time**
   * Measures and displays the time taken to execute a command.
9. **$ cal**
   * Displays the calendar for the current month.
10. **$ clear**
    * Clears the terminal screen.
11. **$ Banner MEET SHAH**
    * Creates a large, banner-like ASCII text with the given name "MEET SHAH" (requires banner package).
12. **$ tty**
    * Prints the file name of the terminal connected to the standard input.

**Practical 3 - Bash Commands Explanation**

**Commands and Explanations**

1. **$ who**
   * Displays logged-in users.
2. **$ who -a**
   * Shows all details about logged-in users and system information.
3. **$ who -H**
   * Displays logged-in users with a header for better readability.
4. **$ who -l -H**
   * Lists active login processes with a header.
5. **$ who -t -H**
   * Displays the last system clock change time with a header.
6. **$ who -m -H**
   * Shows the current terminal session with a header.
7. **$ who -q**
   * Displays the number of users currently logged in.
8. **$ who --boot**
   * Shows the last system boot time.
9. **$ who --ips**
   * Displays the IP addresses of logged-in users.
10. **$ who --lookup**
    * Resolves and displays the hostname for logged-in users.
11. **$ who --runlevel**
    * Displays the current runlevel of the system.
12. **$ who --short**
    * Shows a concise version of logged-in users.
13. **$ who --mesg**
    * Indicates the message permission status for users.
14. **$ who --user**
    * Lists logged-in users and their processes.
15. **$ pwd**
    * Displays the present working directory.
16. **$ sudo -i**
    * Switches to the root user (requires password).
17. **$ logout**
    * Logs out the current user session.
18. **$ whoami**
    * Displays the username of the current user.
19. **$ who; clear; whoami**
    * Shows logged-in users, clears the terminal, then displays the current username.
20. **$ who; tty; date**
    * Displays logged-in users, the terminal name, and the current date.

**Practical 4 - Bash Commands Explanation**

**Commands and Explanations**

1. **cd**
   * Changes the current directory to the user's home directory by default.
2. **pwd**
   * Prints the current working directory.
3. **sudo systemctl stop service-name**
   * Stops the specified service on the system.
   * Replace service-name with the name of the service you want to stop.
   * Example: sudo systemctl stop apache2.
4. **service --status-all**
   * Lists the status of all available services on the system.
   * Displays + for running services and - for stopped ones.
5. **sudo systemctl start service-name**
   * Starts the specified service.
   * Replace service-name with the name of the service you want to start.
   * Example: sudo systemctl start apache2.
6. **sudo systemctl restart service-name**
   * Restarts the specified service.
   * Useful for applying changes to configuration files without stopping and starting manually.
   * Example: sudo systemctl restart apache2.

**Practical 5 - Process Management Commands Explanation**

**Commands and Explanations**

1. **ps**
   * Displays the currently running processes on the system.
   * Example: ps aux shows detailed information about all processes.
2. **wait**
   * Waits for a process to finish.
   * Can be used in a script to pause until a specific process completes.
3. **sleep 30**
   * Pauses the execution for a specified amount of time (in seconds).
   * Example: sleep 30 waits for 30 seconds.
4. **exit**
   * Exits the current shell session or script.
5. **kill**
   * Sends a signal to terminate a process.
   * Typically used with a PID (process ID).
   * Example: kill 1234 terminates the process with PID 1234.
6. **ps -p$$**
   * Shows the process details for the current shell's process.
   * $$ refers to the current shell’s process ID.
7. **kill -l**
   * Lists all available signal names that can be sent with the kill command.
   * Example: kill -l shows all the signal types.
8. **kill pid**
   * Terminates the process with the given PID (process ID).
   * Example: kill 1234 stops the process with PID 1234.
9. **kill -s**
   * Sends a specific signal to a process by name.
   * Example: kill -s TERM 1234 sends the TERM signal to PID 1234.
10. **kill -L**

* Lists all the signals available for kill in a human-readable format.

1. **echo $$**

* Prints the process ID (PID) of the current shell.
* The value of $$ is the PID of the shell running the command.

**Execute file operation**

1. **ls**  
   Lists the contents of a directory.
2. **mv <filename> <destination>**  
   Moves or renames a file or directory.
3. **cp <filename> <destination>**  
   Copies a file or directory.
4. **split <filename>**  
   Splits a file into smaller parts.
5. **join <filename1> <filename2>**  
   Joins lines of two files based on a common field.
6. **rm <filename>**  
   Removes (deletes) a file or directory.
7. **cat <filename>**  
   Concatenates and displays the content of a file.
8. **touch <filename>**  
   Creates an empty file or updates the timestamp of a file.
9. **head <filename>**  
   Displays the first few lines of a file.
10. **tail <filename>**  
    Displays the last few lines of a file.
11. **awk**  
    A powerful text processing tool for pattern scanning and processing.
12. **grep <pattern> <filename>**  
    Searches for a specific pattern in a file.
13. **sort <filename>**  
    Sorts lines of a file.
14. **uniq <filename>**  
    Removes duplicate lines from a file.
15. **less <filename>**  
    Displays a file one screen at a time.
16. **more <filename>**  
    Similar to less, but with limited features.
17. **tr**  
    Translates or deletes characters from a file or input.
18. **pr <filename>**  
    Formats text for printing, used to prepare text for page breaks.
19. **fmt <filename>**  
    Re-formats text to fit within a specified width.
20. **sed <filename>**  
    Stream editor used for parsing and transforming text.
21. **ls -l**  
    Lists directory contents in long format, showing detailed file information.
22. **ls -a**  
    Displays all files, including hidden ones (those starting with a dot).
23. **ls -la**  
    Lists directory contents in long format, including hidden files.
24. **ls -t**  
    Sorts the files by modification time.
25. **ls -S**  
    Sorts the files by file size.
26. **ls \***  
    Lists all subdirectories in the current directory.

**Dir and file manipulation**

1. **mkdir <directory>**  
   Creates a new directory.
2. **cd <directory>**  
   Changes the current working directory to the specified directory.
3. **rmdir <directory>**  
   Removes an empty directory.
4. **pwd**  
   Prints the current working directory.
5. **diff <file1> <file2>**  
   Compares two files line by line.
6. **comm <file1> <file2>**  
   Compares two sorted files line by line and outputs common lines and unique lines.
7. **pr <filename>**  
   Formats text for printing, used to prepare text for page breaks.
8. **dir**  
   Lists the contents of a directory in a format similar to ls.
9. **chmod <permissions> <filename>**  
   Changes the permissions of a file or directory.
10. **cmp <file1> <file2>**  
    Compares two files byte by byte.
11. **ls -l**  
    Lists directory contents in long format, showing detailed file information.
12. **help [option][text-string/pattern/internal-command]**  
    Displays help information for commands or patterns.
13. **tr**  
    Translates or deletes characters from input or files.

**WC,Grep,Tr,Sort,cat,cut,paste**

1. **tr "[:lower:]" "[:upper:]" <filename.txt>**  
   Converts lowercase characters to uppercase in the file.
2. **tr -d "Pune" < filename.txt**  
   Deletes occurrences of the word "Pune" from the file.
3. **cat <filename.txt>**  
   Displays the content of the file.
4. **cut -c 1,10 <filename.txt>**  
   Extracts the 1st and 10th characters from each line in the file.
5. **paste <filename.txt> <filename1.txt>**  
   Merges lines of two files side by side.
6. **wc <filename.txt>**  
   Displays the number of lines, words, and characters in the file.
7. **wc -c <filename.txt>**  
   Counts the number of characters in the file.
8. **wc -l <filename.txt>**  
   Counts the number of lines in the file.
9. **wc -w <filename.txt>**  
   Counts the number of words in the file.
10. **wc --version**  
    Displays the version of the wc command.
11. **spell <filename.txt>**  
    Checks the file for spelling errors.
12. **sort <filename.txt>**  
    Sorts the lines of the file alphabetically.
13. **sort -k 3 <filename.txt>**  
    Sorts the file based on the 3rd column.
14. **grep -c name <filename.txt>**  
    Counts the occurrences of the word "name" in the file.
15. **grep -c name <filename.txt> <filename1.txt>**  
    Counts the occurrences of the word "name" in multiple files.
16. **grep -c name <filename.txt> <filename1.txt> <filename2.txt>**  
    Counts the occurrences of the word "name" in three files.
17. **more <filename.txt>**  
    Displays the content of the file one screen at a time.
18. **cut [-f/-c] [n1,n2] <filename.txt>**  
    Extracts specified fields or characters from each line of the file.