#_ Essential BASH Operations [+150]

1. Basic Commands:

- echo: Print given arguments.
- pwd: Print working directory.
- cd: Change directory.
- ls: List directory contents.
- cat: Concatenate and display file content.
- man: Display manual pages.
- exit: Exit the shell.
- clear: Clear the terminal screen.
- history: Display command history.

2. File Operations:

- touch: Create an empty file.
- cp: Copy files or directories.
- mv: Move/rename files or directories.
- rm: Remove files or directories.
- find: Search for files in a directory hierarchy.
- grep: Search text using patterns.
- head: Output the first part of files.
- tail: Output the last part of files.
- diff: Compare files line by line.
- chmod: Change file permissions.
- chown: Change file owner and group.

3. Process Management:

- ps: Display current processes.
- top: Display dynamic real-time process viewer.
- kill: Send a signal to a process.
- bg: Put processes in the background.
- fg: Bring processes to the foreground.

By: Waleed Mousa

4. Networking:

- `ping`: Send ICMP ECHO_REQUEST packets to network hosts.
- `netstat`: Print network connections, routing tables, statistics, etc.
- `ifconfig`: Display or modify network interfaces.
- `ssh`: OpenSSH client (remote login program).
- `scp`: Securely copy files between hosts.

5. Environment & Variables:

- `env`: Display, set, or remove environment variables.
- `export`: Set an environment variable.
- `unset`: Unset an environment variable.
- `alias`: Create command shortcuts.
- `$PATH`: Environment variable specifying command search path.
- `$HOME`: Environment variable indicating user's home directory.

6. Pipelines & Redirection:

- `|`: Pipe; use output of one command as input to another.
- `>`: Redirect standard output.
- `>>`: Append to file.
- `<`: Redirect standard input.
- `2>`: Redirect error output.
- `2>&1`: Redirect error output to standard output.

7. Text Processing:

- `sort`: Sort lines in text files.
- `cut`: Remove sections from each line of files.
- `awk`: Text processing language.
- `sed`: Stream editor for filtering and transforming text.
- `wc`: Print newline, word, and byte counts for files.

By: Waleed Mousa

8. Script Basics:

- `#! (shebang)`: Denotes script interpreter.
- `variables`: Store data for processing.
- `arrays`: Indexed and associative arrays.
- `if..then..else`: Conditional execution.
- `for`, `while`, `until`: Loop constructs.
- `functions`: Declare modular blocks of code.
- `arguments`: `$1, $2...` for script arguments, `$#` for argument count, `$@` for all arguments as a list.
- `return`: Exit a function.
- `source`: Execute commands from a file in the current shell.

9. Advanced Scripting:

- `case`: Conditional execution based on pattern matching.
- `select`: Generate menus.
- `trap`: Respond to runtime events (signals).
- `string manipulations`: `${#string}`, `${string:position}`, `${string:position:length}` etc.
- `arithmetic`: Use `$((expression))` or `let "expression"`.
- `set`: Change shell options.
- `shift`: Shift positional parameters.

10. Regex & Pattern Matching:

- `*`: Match any string of characters.
- `?`: Match a single character.
- `[...]`: Match any character in the set.
- `[^...]`: Match any character not in the set.

11. Command Line Tricks:

- `ctrl + r`: Search through command history.
- `ctrl + c`: Terminate current command.
- `ctrl + z`: Suspend current command.
- `ctrl + l`: Clear screen.
- `!!`: Execute last command.

By: Waleed Mousa

- `!$`: Last argument of the previous command.
- `!<command>`: Last time `command` was run.
- `{command1,command2}`: Execute multiple commands.

12. Job Control:

- `jobs`: List background jobs.
- `&`: Run command in background.
- `nohup`: Run command immune to hangups.

13. System Info & Monitoring:

- `df`: Report filesystem disk space usage.
- `du`: Estimate file and directory space usage.
- `free`: Display memory usage.
- `uptime`: Show system uptime.
- `w`: Show who is logged on and what they're doing.
- `who`: Display who's on the machine.
- `lsof`: List open files.

14. Archiving & Compression:

- `tar`: Archive files.
- `gzip`: Compress files.
- `gunzip`: Decompress files.
- `zip`: Package and compress files.
- `unzip`: Extract compressed files in ZIP format.

15. Package Management (specific to distribution):

- `apt-get`: APT package handling utility (Debian).
- `yum`: Package manager (RedHat, CentOS).
- `dnf`: Next-generation package manager (Fedora).
- `pacman`: Package manager (Arch Linux).

16. Permissions & Ownership:

- `chgrp`: Change group ownership.
- `chown`: Change file owner and group.

By: Waleed Mousa

- `umask`: Set default permissions.

17. Disk Usage:

- `fdisk`: Manipulate disk partition table.
- `mkfs`: Create filesystem.
- `mount`: Mount a filesystem.
- `umount`: Unmount a filesystem.

18. Bash Options:

- `set -e`: Exit on error.
- `set -u`: Treat unset variables as errors.
- `set -x`: Print commands before executing them.

19. Text Editors in Bash:

- `vi`/`vim`: Opens the Vim (or Vi) text editor.
- `nano`: Opens the Nano text editor.
- `emacs`: Opens the Emacs text editor.

20. Bash Shortcuts & Keybindings:

- `ctrl + a`: Move to the beginning of the line.
- `ctrl + e`: Move to the end of the line.
- `ctrl + u`: Delete from cursor to the beginning of the line.
- `ctrl + k`: Delete from cursor to the end of the line.
- `ctrl + w`: Delete from cursor to the beginning of the word.
- `ctrl + y`: Paste the last deleted command.
- `ctrl + t`: Swap the last two characters before the cursor.

21. Bash Special Variables:

- `$?`: Holds the exit status of the last command executed.
- `$$`: Holds the process ID of the current script.
- `$0`: The name of the script itself.
- `$*`: All the arguments are double quoted. If a script receives two arguments, `$*` is equivalent to `$1 $2`.
- `$@`: All the arguments are individually double quoted.

By: Waleed Mousa

22. Command Substitution & Evaluation:

- `$(command)`: Execute `command` and replace it with its output.
- `` `command` ``: (backticks) Another way to execute `command` and replace it with its output.
- `$((expression))`: Evaluate arithmetic expression.

23. Brace Expansion:

- `{a,b,c}`: Generates the strings 'a', 'b', and 'c'.
- `{0..5}`: Generates the strings '0', '1', '2', '3', '4', '5'.

24. Wildcard Patterns:

- `*`: Matches any string, including the null string.
- `?`: Matches any single character.
- `[class]`: Matches any character in the given class.

25. String Manipulation in Bash:

- `${variable#pattern}`: Remove from `$variable` the shortest part of `$pattern` that matches the front end of `$variable`.
- `${variable##pattern}`: Remove from `$variable` the longest part of `$pattern` that matches the front end of `$variable`.
- `${variable%pattern}`: Remove from `$variable` the shortest part of `$pattern` that matches the back end of `$variable`.
- `${variable%%pattern}`: Remove from `$variable` the longest part of `$pattern` that matches the back end of `$variable`.

26. I/O Redirection & File Descriptors:

- `1>` or `>`: Redirects standard output.
- `2>`: Redirects standard error.
- `&>`: Redirects both standard output and error.
- `1<&-`: Closes standard input.
- `2<&-`: Closes standard error.

By: Waleed Mousa

27. Advanced Commands:

- `xargs`: Read items from standard input, then execute a command with those items.
- `seq`: Generate a sequence of numbers.
- `watch`: Execute a program periodically, showing output fullscreen.
- `date`: Display or set the system date and time.

28. Networking & Analysis:

- `curl`: Transfer data from or to a server.
- `wget`: Retrieve files using HTTP, HTTPS, and FTP.
- `dig`: Query DNS name servers.
- `traceroute`: Display the route packets take to a network host.

29. Permission & Identity:

- `sudo`: Execute a command as another user.
- `su`: Change user ID or become a superuser.
- `id`: Display user identity.

30. Debugging & Bash Options:

- `set -v`: Print shell input lines as they're read.
- `set -n`: Read commands but do not execute them.
- `set -x`: Print commands and their arguments as they're executed.

By: Waleed Mousa