Basic Commands 2

Practice 1: Using ping

- **Objective**: Check the connectivity between your machine and a remote server.
- **Task**: Use the ping command to check the connectivity to www.google.com. Record the number of packets sent, received, and the packet loss percentage.

Practice 2: Using man

- **Objective**: Learn how to use the manual pages to find information about commands.
- **Task**: Use the man command to find the manual entry for grep. Summarize the options -i, -v, and -c.

Practice 3: Using cal and jcal

- **Objective**: Display calendars.
- **Task**: Use the cal command to display the calendar for the current month. Use the jcal command to display the Julian calendar for the current month.

Practice 4: Using who and whoami

- **Objective**: Identify users logged into the system and the current user.
- **Task**: Use the who command to list all users currently logged into the system. Use the whoami command to display your username.

Basic Filters

Practice 5: Using grep

- **Objective**: Search for specific patterns in a text file.
- **Task**: Create a file named sample.txt with multiple lines of text. Use grep to find all lines containing the word "Linux".

Practice 6: Using tee

- **Objective**: Redirect output to a file and display it on the terminal.
- **Task**: Use the ls command to list directory contents, and pipe the output to tee to save it to dir_list.txt while also displaying it on the terminal.

Practice 7: Using tr

- **Objective**: Translate or delete characters.
- **Task**: Use the tr command to replace all spaces in sample.txt with underscores.

Practice 8: Using uniq

- **Objective**: Report or filter out repeated lines in a file.
- **Task**: Create a file named duplicates.txt with some duplicate lines. Use uniq to filter out duplicate lines.

Practice 9: Using sort

- **Objective**: Sort lines of text files.
- **Task**: Use the sort command to sort the lines in sample.txt alphabetically.

Practice 10: Using wc

- **Objective**: Count lines, words, and characters in a file.
- **Task**: Use the wc command to count the number of lines, words, and characters in sample.txt.

File Compression

Practice 11: Compressing Files

- **Objective**: Learn to compress and decompress files.
- **Task**: Create a text file named data.txt. Compress it using gzip and then decompress it.

Practice 12: Creating and Extracting Archives

- **Objective**: Learn to create and extract tar archives.
- **Task**: Create a directory named backup with some files. Use tar to create an archive named backup.tar.gz. Extract the contents of the archive to verify it.

File Security

Practice 13: Changing File Permissions

- **Objective**: Modify file permissions using chmod.
- **Task**: Create a file named secure.txt. Change its permissions to allow only the owner to read and write to it.

Practice 14: Changing File Ownership

- **Objective**: Change the owner and group of a file.
- **Task**: Use the chown command to change the owner and group of secure.txt to another user and group on the system (you may need superuser privileges).

Bash Scripting

Practice 15: Writing a Simple Bash Script

- **Objective**: Create and run a basic bash script.
- **Task**: Write a bash script named hello.sh that prints "Hello, World!" to the terminal. Make the script executable and run it.

Practice 16: Using Variables in Bash Scripts

- **Objective**: Work with variables in a bash script.
- **Task**: Write a bash script named greet.sh that takes a user's name as an argument and prints "Hello, [name]!"

Practice 17: Conditional Statements in Bash Scripts

- **Objective**: Use if-else statements in a bash script.
- **Task**: Write a bash script named check_number.sh that takes a number as an argument and prints whether the number is positive, negative, or zero.

Practice 18: Loops in Bash Scripts

- **Objective**: Use loops in a bash script.
- **Task**: Write a bash script named countdown.sh that counts down from 10 to 1 and then prints "Liftoff!"

Practice 19: Using Functions in Bash Scripts

- **Objective**: Define and use functions in a bash script.
- **Task**: Write a bash script named math.sh that defines two functions: one for adding two numbers and one for subtracting two numbers. The script should take three arguments: the operation (add or subtract) and the two numbers. Call the appropriate function based on the operation.

Practice 20: Reading User Input in Bash Scripts

- **Objective**: Prompt the user for input and use it in the script.
- **Task**: Write a bash script named user_input.sh that prompts the user for their name and age, and then prints a message saying "Hello, [name]! You are [age] years old."

Practice 21: Using Arrays in Bash Scripts

- **Objective**: Create and manipulate arrays in a bash script.
- **Task**: Write a bash script named array_example.sh that creates an array of five names, prints each name, and then prints the total number of names in the array.

Practice 22: Working with Files in Bash Scripts

- **Objective**: Perform file operations in a bash script.
- **Task**: Write a bash script named file_operations.sh that takes a directory path as an argument, lists all files in the directory, and counts the number of files. If the directory does not exist, the script should print an error message.

Practice 23: Creating a Daily Backup Script

- **Objective**: Automate the process of backing up files daily.
- **Task**: Write a bash script named daily_backup.sh that creates a backup of a specified directory and saves it with a timestamp. Set up a cron job to run this script every day.