Here are some **intermediate-level Linux tasks** to practice and enhance your command-line skills:

**File and Directory Management**

1. Find all files larger than 10 MB in the /home directory.
2. Create a directory structure like projects/{project1,project2}/{src,bin,docs} in a single command.
3. Copy all .log files from one directory to another while preserving their timestamps.
4. Create a symbolic link named shortcut pointing to a file of your choice.
5. Compress the projects directory into a .tar.gz archive.

**File Permissions and Ownership**

1. Change the ownership of all files in the projects directory to a user named developer.
2. Grant execute permission to all .sh files in a directory.
3. Set the sticky bit on a directory to prevent others from deleting files they don’t own.
4. Find files in the /var directory that are writable by the group.

**Text Processing**

1. Extract the 3rd column from a CSV file named data.csv.
2. Count the occurrences of the word "error" in a log file.
3. Remove all empty lines from a file named report.txt.
4. Sort a file named scores.txt by the second column.
5. Extract lines between the 10th and 20th line in a file named log.txt.

**Process and System Monitoring**

1. Monitor real-time system resource usage (CPU, memory, disk I/O) for 1 minute.
2. Kill all processes running a specific program, such as apache2.
3. Display the top 5 memory-consuming processes on your system.
4. Check which processes are using the most disk I/O.
5. Display the number of threads used by the ssh service.

**Networking**

1. List all active network connections on your system.
2. Download a file from the internet using wget or curl.
3. Find the open ports on your system using netstat or ss.
4. Test the speed of your internet connection using a command-line tool.
5. Display the route that packets take to reach a website (e.g., google.com).

**Shell Scripting**

1. Write a script that takes a filename as input and displays whether it exists and its size.
2. Create a backup script that compresses and archives a directory daily.
3. Write a script to check if a service (e.g., nginx) is running, and restart it if not.
4. Automate the deletion of files older than 7 days in a directory.
5. Write a script to monitor disk usage and send an email alert when it exceeds 80%.

**System Configuration**

1. Add a cron job that clears temporary files every week.
2. Configure a user account with limited disk space using quotas.
3. Change the default shell for a user to /bin/zsh.
4. Set up SSH key-based authentication for a remote server.
5. Mount a USB drive manually and unmount it afterward.

**Archiving and Compression**

1. Compress multiple files into a .zip archive.
2. Extract specific files from a .tar.gz archive without extracting the whole archive.
3. Split a large file into smaller chunks of 100 MB each.
4. Merge the chunks back into the original file.

**User and Group Management**

1. Create a new group named admins and add a user to it.
2. Create a user account that expires after 30 days.
3. Lock and unlock a user account.
4. List all currently logged-in users and their activity.

**Advanced Search and Filters**

1. Find all files in /var/log modified in the last 24 hours.
2. Search for a specific IP address in a log file and display the surrounding 5 lines.
3. Replace all occurrences of "http" with "https" in a file named urls.txt.
4. Identify and delete duplicate lines in a file.